October 1985 U.S. $3 Canada $3.95

Just for your Sanyo MBC-550, 555 and 775

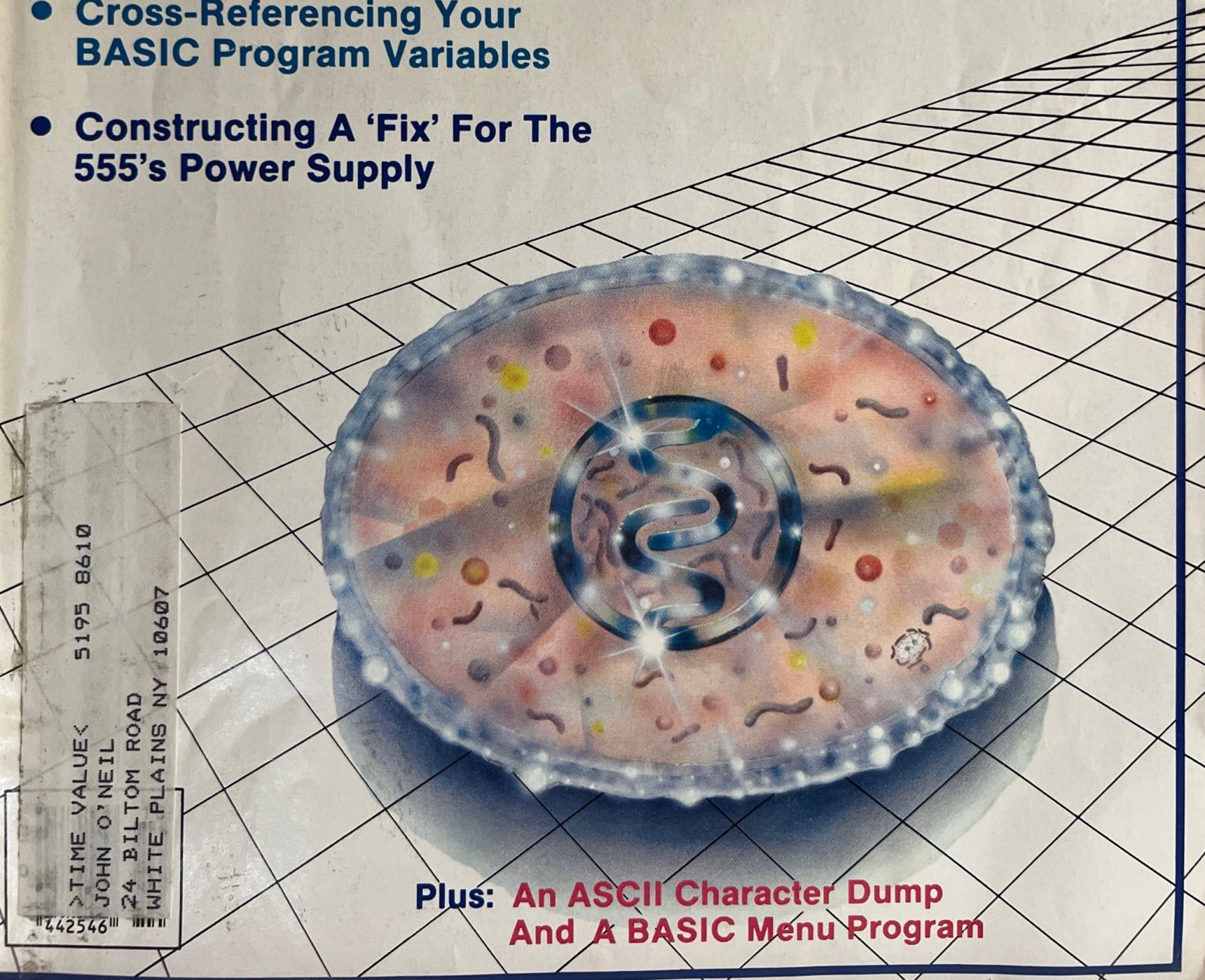
# soft sector®

The Monthly Magazine For Sanyo Personal Computer Users

# Life On The Sanyo

## A New Look At An Old Favorite

- **Creating Graphic Screen Dumps With A Daisy Wheel Printer**
- **Cross-Referencing Your BASIC Program Variables**
- **Constructing A 'Fix' For The 555's Power Supply**



**Plus: An ASCII Character Dump And A BASIC Menu Program**

# PICASSO

## The **ULTIMATE** In Computer Graphics Programs


**PICASSO** by Bill Dunlevy

This is the one to set the standards! PICASSO does more than many graphics programs costing thousands of dollars more. Never before has such power and simplicity been available to the user.

What does PICASSO do? Everything!

* Large, helpful picture menus, eliminate the need to memorize control keys or cumbersome graphic languages.
* Keyboard, Joystick, or even Graphic Tablet controlled, in "directional" or "free-hand" format.
* You can choose from 64 pre-mixed colors, or mix your own for over 10,000 possible patterns, and save your favorites to a palette on disk!
* Over 20 different "brushes" let your draw with symbols in different "densities" for an air-brushed effect.
* Special "Rubber band" graphics let you see what you're doing AS you make adjustments on lines, rays, boxes, circles, polygons, stamps, etc.
* Versatile "Fill" permits you to change multi-color patterns.
* You're not restricted to simple circles and elipses here! You can create ANY kind of polygon: from triangles, to eliptical pentagons, to perfect circles and everything in between!
* Powerful "Stamp" system lets you copy, move, save, and perform complexed operations with any sections of your pictures.
* You can use your picture files with MichTron's new FREEZE FRAME graphic printing utility for fantastic hardcopy pictures, and in your own BASIC or machine language programs for all your graphic needs.

What DOESN'T PICASSO do? Restrict your imagination.

PICASSO is guaranteed to knock your smocks off!

256K SANYO MBC-550/555 required

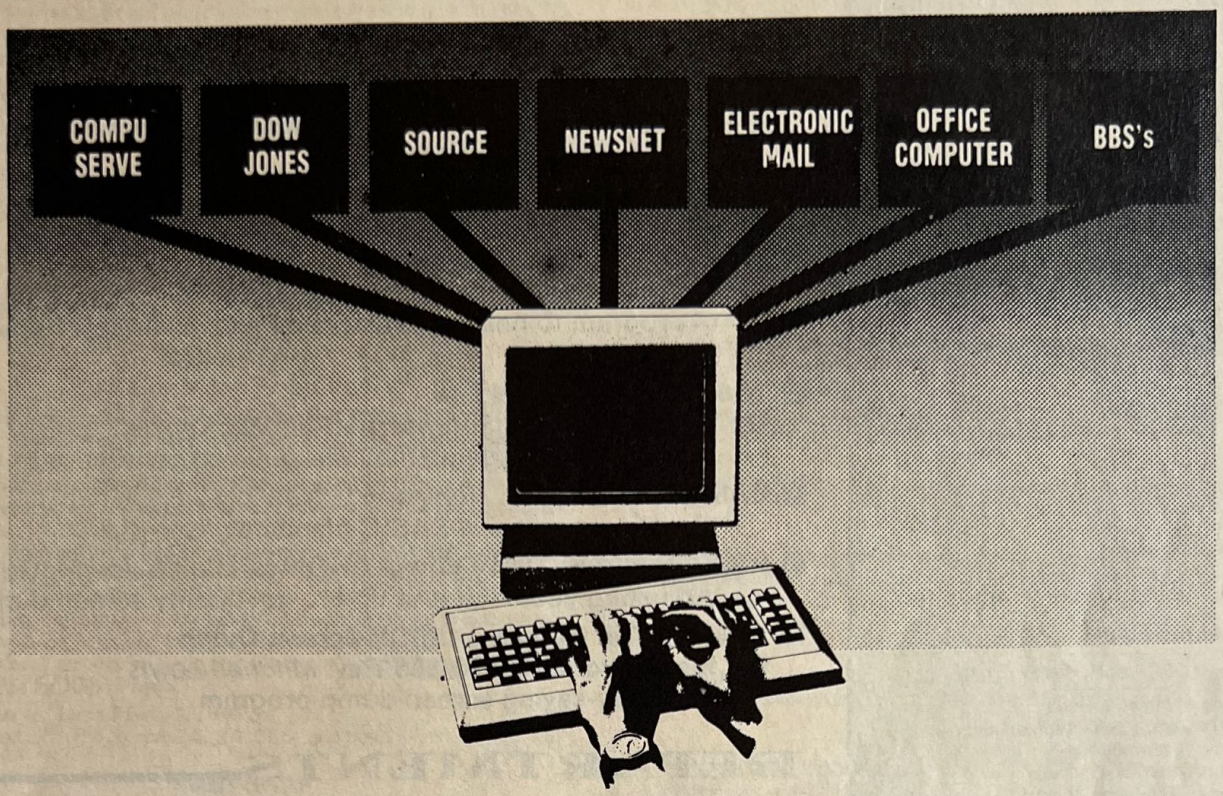
(Joystick recommended) . . . . . **$99.95**

MichTron

576 S. Telegraph Road Pontiac, Michigan 48053

Orders & Info: (313) 334-5700

# INTELLICOM




## A COMPLETE Communications System

Now you can tap any of the countless telephone data based INFO sources, EXTRACT an FILE that data away for use later. Suddenly, your computer is the smartest terminal around because INTELLICOM emulates such terminals as:

- **DigitalVT52/VT100**
- **TeleVideo 910 • ADM3A/5**
- **ADDS Viewpoint/25**
- **CompuServe Executive**
- **Simple TTY**
- **TeleVideo 925**

Now you can transfer **both binary and text** files, using simple ASCII, ASCII Capture, Xon/Xoff, XMODEM (both Cheksum and CRC are supported) or CompuServe A Protocols. Operate at any speed up to 4800 (Sanyo) or 9600 baud (IBM PC) without ever being outdated as higher speed/lower priced modems are introduced.

Simple menu selections, with a built in **HELP** facility keep it easy forever.

And our expert mode will allow you to even bypass the menus. **Autodial directories (Hayes or equivalent modems) are supported along with simultaneous printing.** Our **built in DOS WINDOW** will actually **let you run another program** from within INTELICOM **(192K min. ram).**

**Additional features: Variable Buffer Size** reduces disk activity, while utilizing RAM capacity. BATch file utility program included. **Persistence feature for autodialing.**

**Auto log-on script files** can be used to support almost any smart modem and/or automatically log into your favorite remote system. **User defined function keys. Color support. Michtron and AOK DOS support** and more! Best of all, the entire versatile INTELLICOM package costs you only **$89.95 COMPLETE! RUSH $15 today and you'll receive a fully functional demo disk that lets you check it all out.** If you order INTELICOM, simply return the demo disk and we'll deduct your $15 from the $89.95 purchase price.

For $10.00 more we will supply you with the host support code for the error checking protocol on Dec system 10, 20, or VAX ... and save your company a bundle!

**COMPUTER TOOLBOX inc.**

MasterCard/VISA/COD orders accepted. Specify computer and DOS version when ordering. Connecticut residents add 7½% sales tax. **Add $5 shipping for all orders.**

**1325 East Main Street, Waterbury, CT 06705**
**In Connecticut call: 597-0273**

# soft sector

**The Monthly Magazine for Sanyo Personal Computer Users**

Vol. II, Issue 3 October 1985

**Editor and Publisher**
Lawrence C. Falk

**Managing Editor** Belinda C. Kirby
**Consulting Editors** Gordon Monnier, T. Kevin Nickols
**Submissions Editor** Jutta Kapfhammer
**Reviews Editor** Monica Dorth
**Editorial Consultants** Tamara Dunn, Danny Humphress
**Contributing Editors** Fred Blechman, Michael W. Ecker, Ph.D., Tim Purves, Brian M. Stone, Charlotte A. Stone
**Technical Assistant** Ed Ellers
**Editorial Assistants** Wendy Falk, Judi Hutchinson, Shirley Morgan

**Art Director** Kevin Quiggins
**Design Staff** Tracey Jones, Heidi Maxedon, Jerry McKiernan

**Chief of Typography** Debbie Hartley
**Typography Services**
Jody Doyle, Angela Kapfhammer, Suzanne Benish Kurowsky

**FPSS, Ag. Publications Enterprises, Inc.**

**President** Lawrence C. Falk

**General Manager** Patricia H. Hirsch

**Editorial Director** James E. Reed
**Asst. Editorial Director** Jutta Kapfhammer
**Executive Editor** Courtney Noe
**Creative Director** Sally A. Gellhaus
**Manager of Public Relations** Wayne Fowler

**Administrative Assistant to the Publisher**
Judy Brashear
**Asst. General Manager for Finance** Donna Shuck
**Bookkeeper** Diane Moore
**Advertising Accounts** Bev Taylor
**Dealer Accounts** Judy Quashnock
**Accounts Payable/ADP** Lisa Ragan
**Director of Fulfillment Services** Bonnie Shepard
**Asst. Dir. of Fulfillment Services** Sandy Apple
**Customer Service Representative** Deidra Henry
**Soft Sector on Disk Subscriptions** Monica Wheat
**Word Processor Manager** Patricia Eaton
**Chief of Production Services** Melba Smith
**Dispatch** Janice Eastburn
**Research Assistants** Beverly Bearden, Laurie Falk, Laila Masri, Sharon Smith

**Advertising Coordinator** Doris Taylor
**Advertising Assistant** Debbie Baxter
**(502) 228-4492**

**For Soft Sector Advertising and Marketing office information, see Page 62.**

**SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users** (ISSN 8755-7460) is published every month of the year by FPSS, Ag. Publications Enterprises, Inc., 9529 U.S. Highway 42, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users, SOFT SECTOR ON DISK** and the **SOFT SECTOR** logotypes are registered ® trademarks of FPSS, Ag.

Second Class Postage Rates are paid at Prospect, Kentucky and additional offices. USPS 741-750. POSTMASTER: Send address changes to SOFT SECTOR, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.





Subscriptions to **SOFT SECTOR — The Monthly Magazine for Sanyo Personal Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.


# CONTENTS

## FEATURES



## DEPARTMENTS



## REVIEWS





■ The small disk symbols appearing beside features and regular columns indicate that the program listings with those articles are on this month's SOFT SECTOR ON DISK, ready to LOAD and RUN. For full details, see the SOFT SECTOR ON DISK ad on Page 55.

# A-OK Computers

## The A-OK Board™—The Ultimate Upgrade™
### "960K + 8 MHz"

**The A-OK Board** lets you double your processor speed and nearly quadruple your RAM. **The A-OK Board** plugs inside your Sanyo with no soldering. **The A-OK Board** works fine with or without the Video Board and is fully compatible with all software which runs on your Sanyo. **The A-OK Board**'s 8088-2 chip does all your processing, even Ramdisk, twice as fast. Plug one bank of 256K chips into **The A-OK Board** to double your RAM to 512K or two banks for 768K or a third bank for 960K of contiguous RAM. See our **TurboDrive550** ad about 800K ramdisks. Options include a real-time clock/calendar. The **A-OK Board** has been completely redesigned so that you can replace the 8MHz 8088-2 with the new 10 MHz or 12 MHz 8088's when they become available. Sorry about past delays, but we didn't want to sell you something that we would soon make obsolete. The **A-OK Board** *is* **The Ultimate Upgrade.**

**A-OK Board** with no memory . . . . . $150
8088-2 chip for high speed on **A-OK Board** . . . . . $ 25
**A-OK Board** with the works . . . . . $395

## TurboLinker™
### "The Linker for Turbo Pascal"

**TurboLinker** accelerates the compilation of your Turbo Pascal 2.0 and 3.0 programs. **TurboLinker** provides a program that turns your debugged and compiled Turbo Pascal procedures into compact, linkable external procedures. Because they are already compiled, these procedures link instantly into the code you are debugging. **TurboLinker**'s linkable procedures use much less memory than the Turbo Pascal source code they come from. Thus you can bring them in off disk faster and store more of them on RAM disk. **TurboLinker** is easy to use, self-documenting, and works from the Turbo Pascal Program Development Environment. An extra bonus—now you can sell procedures written in Turbo Pascal as linkable procedures without having to disclose your source code. A-OK is looking for high-quality Turbo Pascal libraries to market. Why waste time recompiling code that you know works? **TurboLinker** is the ideal upgrade for Turbo Pascal.

**TurboLinker** for Turbo Pascal 2.0 and 3.0 . . . . . $49
**TurboLinker** for Turbo Pascal 2.0, 3.0, BCD, and 8087 . . . . . $75

## SANYGRAF
### "Professional Business Graphics"

A-OK searched the world to bring you the best business graphics for your Sanyo. We are delighted by what we found—**SANYGRAF. SANYGRAF**'s well thought-out menus and superb documentation, make this extremely versatile software very easy to master. **SANYGRAF** is *very* fast and even faster with an 8087. **SANYGRAF** supports popular plotters. **SANYGRAF** is so good that Sanyo Australia supplies it with every color graphics system it sells. 2.11, 256K, one DSDD or DSQD drive required.

**SANYGRAF** . . . . . $99

## TurboDrive550™
### "A Quality Ram Disk for Sanyo"

**TurboDrive550** turns your unused RAM into a disk drive.
**TurboDrive550** is very fast.
- Speeds up all disk intensive software.
- Makes WordStar and FORMSORT run more quickly.
  - Accesses disk 75,000 times faster than hard disk.
- Makes 800K ramdisks on the **A-OK Board**.
  - Using the new **A-OK DOS-3** with your single or double-sided drives.
  - Using the new **A-OK DOS-4** with your quad drives.

**TurboDrive550** comes **FREE** with every new **A-OK DOS 1-4™.**
**TurboDrive550** for other Sanyo DOSs . . . . . $33
(The best deal is to buy **A-OK DOS-1™** for $35 and get **TurboDrive550** free.)

## Microsoft FORTRAN 77 (limited supply) . . . . . $219

## VidSwitch550™

Just push a button on the left front side of your Sanyo and you've switched from the Sanyo graphics port to the Video Board port. Another push of the button on this small box switches it back. The monochrome version requires only the mono cable you already have. The color version requires no extra cables. Everything just plugs in.

**VidSwitch550**—Monochrome . . . . . $29
**VidSwitch550**—Color (with sync inversion) . . . . . $69

## XCord550™

6 foot extension cord for your Sanyo PC keyboard. . . . . $15

## RS232C Serial Boards . . . . . $72

## The A-OK™ DOS Family
"Upgrade your MS-DOS 1.25 or 2.11 and Video Board"

All members of the **A-OK DOS** family add these features to your MS-DOS:
- **A-OK DOS** supports RAM expansion to 960K.

With the Video Board, **A-OK DOS** uses 24K *less* memory than the "competition". With extended memory, **A-OK DOS** uses 55K *less* memory than the "competition".
- **A-OK DOS** now includes *free* high-quality ramdisk software—**TurboDrive550** (a $33 value).
- Still the only text dump for 2.11 automatically transferred via FORMAT /S.
- User-controlled scrolling in all subsystems, not just DOS.
- **A-OK DOS** system messages are easier to understand.
- Supports up to 4 disk drives.

**A-OK DOS** versions for MS-DOS 2.11:
- Run all software which runs on the Sanyo under MS-DOS 2.11 and Video Board, and support installable drivers for hard disks, etc.
- Supports the only graphics screen dump software built into DOS—**PrintScreen550.**

*Soft Sector* sums it up well in their review entitled "A-OK DOS and PrintScreen550—An Unbeatable Combination."
Comparing **A-OK DOS-3** and DS DOS+, the *Sanyo PC Hackers Newsletter* states "In a nutshell, you get more for your money in A-OK's package".
A-OK has tailored different versions of its DOS for varying disk drive configurations. This makes our ad more complex, but our DOS more efficient and easier for you to use.

## A-OK DOS-1™
- Any combination of up to four single and double-sided drives.
- Double-sided drives support single and double-sided diskettes.
- 8 and 9 sector formats—360K, 320K, 180K, and 160K . . . . . **$35**

## A-OK DOS-2™
- All the features and formats of **A-OK DOS-1**
- PLUS it reads, writes, and formats 400K DSDD formats . . . . . **$45**

## A-OK DOS-3™
"**A-OK DOS** for Double and Quad-Density Drive Systems"
- All the features of **A-OK DOS-2**
- PLUS up to 4 double-density and quad-density drives
  - on double-density drives:
    - reads and writes all formats read by 1.25 and 2.11
  - on quad-density drives:
    - reads, writes, and formats 730K and 800K on ordinary diskettes.

**A-OK DOS-3**/1.25 or **A-OK DOS-3**/2.11 and Video Board . . . . . **$55**

## A-OK DOS-4™
### "The *Standard* Quad-Density DOS"
- All the features of **A-OK DOS-2**
- PLUS up to 4 quad drives—3.2 Megabytes of *removable* medium!
  - quad drives read all diskettes written by MS-DOS 1.25 and 2.11
  - quad drives read, write and format 160K, 180K, 200K, 320K, 360K, 400K, 730K, 800K formats on ordinary diskettes.
  - will not run Prolok™ disks.

Scottsdale Systems, America's No. 1 seller of Sanyos, selected **DOS-4** as the standard DOS supplied with their Silver Fox™. (HAGEN-DOS™ is A-OK DOS-4).
- Unique support for Lotus 1-2-3™ and Picasso™ on quad drives.
- FORMATs, READs, WRITEs, and DISKCOPYs *all* MS-DOS standard formats employed by the Sanyo 550 Series.

**A-OK DOS-4** 1.25 or **A-OK DOS** 2.11 and Video Board . . . . . **$55**

## PrintScreen550™
"Fast, High-Resolution Graphics and Text from Screen to Printer with One Keystroke"
- **PrintScreen550** now also supports standard Sanyo graphics and the Video Graphics Board.
- Prints a screenful of graphics to most popular printers.
- User has the option to rotate the image on the paper.
- Multiscreen facility permits a printed image to be 640 dots by an almost unlimited number of dots in the perpendicular direction.

**PrintScreen550** is an option to any A-OK DOS for a meager . . . . . **$24**

## Orders may be placed immediately.

Please send money order or cashiers check. Personal or company checks require 2-3 weeks to clear. Prices reflect a 3% cash discount. MasterCard/VISA accepted. MD residents add 5% sales tax. Out-of-state order, no tax. Prices subject to change. All brands are registered trademarks. 20% restocking fee for all returned merchandise. No returns accepted without a Return Autthorization Number.

**A-OK Computers 816 Easley St., Suite 615 • Silver Spring, MD 20910 301-585-5105, 301-585-5106**


# LETTERS TO THE EDITOR

PAR AVION

## PARTS IS PARTS

*Editor:*

We at St. Mary's Church are long time users of Sanyo computers, and we could not exist without your magazine. The information we obtain from it just cannot be measured.

Take for example your article "Making the Switch To The Video Board" (SOFT SECTOR, June '85, Page 17), which is a very helpful bit of information. I have installed the switch and it works extremely well. You might want, however, to correct one part of the article.

One of the parts needed there was listed one #15-1539 VCR Video/Audio Cable(s). Radio Shack does not have this part anymore, and I purchased two #42-2368, 3.6, 12-inch Shielded HI FI Cables instead, at $3.09 each which have worked out perfectly.

I am sure your readers will be very happy with this set up for monochrome monitors.

*Bud Buescher*
*Parish Secretary*
*Bloomington, IL*

---

## DOUBLE DRIVE

*Editor:*

I recently attempted to upgrade my MBC-550 by adding a second disk drive. The second drive is identical to the original drive, a TEAC Model FD-54A-00-U 160K single-sided. After following the instructions in the *Sanyo Operator's Guide* to the letter, I was more than a little disappointed to find that the drives would not work satisfactorily. Obviously some vital instruction is missing from the handbook because when connected separately, both drives work perfectly, but when both are connected at the same time, both drives run simultaneously. Upon studying the plugs to the two drives, it appears that they are connected in parallel so perhaps the results I am getting are not really so surprising. Rather than begin experimenting and risk wrecking both drives, I would be most grateful if you would let me into the secret.

May I offer my congratulations on an excellent publication which I read from cover to cover. The prices of the articles in the advertisements almost make me weep; about one third of what we must pay here in New Zealand.

Thanking you in anticipation, and very best wishes for the future of SOFT SECTOR.

*G. Lacey*
*Masterton, New Zealand*

*Editor's Note:*
*This question was sent to John Kelty, author of "TEAC FD-54A LED Modification" (SOFT SECTOR, April '85, Page 88), for a reply.*

*Dear G. Lacey:*

I received your letter from the SOFT SECTOR managing editor. After reading it about halfway through, it occurred to me that your problem with the disk drives may be too simple. If you followed the Sanyo manual to the letter, which it seems that you did, it does not mention the drive select jumpers. You must use DS0 for drive A and DS1 for drive B. The clue in your letter that suggests this is your problem, is when you state that "either drive works by itself." If one of the drives is selected as DS1 (drive B), then you should not be able to boot up the system with this drive. The diskette will rotate, but the light should not come on and no data should be read from it.

You will also notice that there are DS2 and DS3 selection jumpers. These are for drive C and drive D if you ever add any external drives to your Sanyo. I hope that this has solved your problems. Please contact me or SOFT SECTOR if you continue to have problems with upgrading your system drives.

*John R. Kelty*
*Lincoln, NE*

---

## SHOW AND TELL

*Editor:*

This letter is in response to a letter which ran in SOFT SECTOR September '85, pertaining to my article "Putting The Hidden Files Where You Can Find Them." The reader stated that it was misleading to say that once a hidden file is made visible, it can be loaded into DEBUG. What I said was, "once a hidden file is made visible, it can be loaded into DEBUG and modified," and by modified I mean the file on disk.

Actually, SHOW.COM does more than just make the files visible. The attribute byte for IO.SYS and MSDOS.SYS in the disk directory contains a 7, which means the three least significant bits are on and indicates that they are read only, hidden, and system files. Attempting to write to these files will only result in an error message. SHOW.COM changes the attribute byte to 0 and the files

can now be loaded into DEBUG, modified, and the changes written to disk. HIDE.COM will then change the attribute byte back to 7.

*Ray C. Robinson*
*Lake Charles, LA*

---

## REINSTATING COLOR

*Editor:*

I tried the *BlitzStar* (SOFT SECTOR, August '85, Page 44) patch in combination with a RAM Disk and was pleasurably amazed at the speed of updating the screen. However, Mr. Annis did not mention that one would lose the reverse video format on the messages. This is a small price to pay, but it does make blocking more difficult. As I would like to have my cake and eat it too, perhaps Mr. Annis could suggest a way of restoring the reverse video messages without sacrificing speed.

*Paul Fuchs*
*Flanders, NJ*

*Editor's Note:*
*See below.*

*Editor:*

The *WordStar* patch contained in the program *BlitzStar* by Charles Annis (August '85, Page 44) is a real time saver. However, I was disappointed to see that the patched program displays both the menu and text in the same color. Not satisfied with a monochrome *WordStar*, I did a little disassembling of my own. I found a *WordStar* internal flag at Location 2B3 Hex which, when changed to FF Hex restores the multicolor display. Furthermore, the menu colors are stored in 284 Hex and the text colors are in 28B Hex. In each of these offsets, the upper four bits represent the background color and the lower four bits represent the foreground color. By changing these manually, Video RAM Board users can take advantage of all of the board's 16 colors.

*Herb Martello*
*Whiteford, MD*

---

## A SYSTEM SWITCH

*Editor:*

Recently, I purchased *Media Master* from DG/Systems in Woodland Hills, CA to convert several thousand lines of PASCAL source code for use on the MBC-555 from a Kaypro II. The program does not work 100 percent effectively, but I did discover a way around this which does allow text file transfer to work. I thought I would pass it on for anyone else needing to bring text files from a CP/M system to the MBC-555.

First, format a system disk on the MBC-555 with the format utility command:

```
FORMAT B:/1/S
```

Use the formatted disk as the destination disk on the CP/M system. Transfer a single file at a time from the CP/M disk to the MS-DOS disk. Do not use a wildcard for the copy as it seems to trash the directory in the process.

*John L. Bluma, CDP*
*Meta, MO*

---

## LINE EDITING

*Editor:*

When writing a program in Sanyo BASIC Version 1.31, there is often a need to use the same or similar line construction in more than one place in the program. In the past, I was forced to re-input the line each time it was needed. The Sanyo screen editor provides a much easier way which does not seem to be documented in the book. Start by typing the line in once:

```
10 PRINT "THIS IS A TEST."
```

Hit the up arrow to get into the screen editor. Edit the line number from 10 to 15.

```
15 PRINT "THIS IS A TEST."
```

Hit BREAK and then LIST. When the program is listed, it will read:

```
10 PRINT "THIS IS A TEST."
15 PRINT "THIS IS A TEST."
```

The screen editor edits and copies your original line to the new location but does not remove the old line. Both of the lines can be edited as normal or reduplicated by editing the line numbers again. If there are intervening lines in the program the duplicated line will appear in its proper numerical order.

I hope that you find this "Helpful Hint" as useful as I do.

*John R. Duchek, Ph.D.*
*St. Louis, MO*

---

## NEW FORMATIONS

*Editor:*

I would like to inform anyone interested, that the Louisville, Ky. area has a Sanyo Users Group. Anyone wanting more information or wanting to join may write or call: Sanyo Users Group, 1616 Utica-Sellersburg Rd., Jeffersonville, IN 47130, (812) 282-2866.

*Charles A. Haven*
*Jeffersonville, IN*

*Editor:*

We are constantly adding other Sanyo SUGs to our mailing list in hopes of exchanging newsletters, etc.

Would you please note our address change. Our club was formerly listed as: Sanyos of Sarasota, Ernie Bontrager, 4047 Bee Ridge Road, Sarasota, FL 33582. Ernie is the owner of the local Sanyo dealership and started the club, but now it is operated independently of the store. Our new club name and address is: Sanyos of SaraMana (S.O.S.), 4411-100th Street West, Bradenton, FL 33507-1619.

Thanks for your help and sharing with us.

*Bill Harrier*
*President*
*Bradenton, FL*

*Editor:*

The MBC-550/555 Sanyo Club of France, has the pleasure of announcing our recent creation.

We have already gotten together several users of this machine. Our ambitions are numerous and diverse. We are trying to develop a system of hardware and software for this machine and get the most out of it that we can. Our address is: Sanyo-Club, 1, Rue de Clementville, 34000 Montpellier, France

*Philippe Chardon*
*President*
*Montpellier, France*

*Editor:*

We have finally managed to get a Sanyo Users Group started in the Cincinnati area. Please add our name to your list of active Sanyo User Groups. Cincinnati Sanyo User Group, Tim Mullen, 336 Miami Valley Drive, Loveland, Ohio 45140, (513) 831-6799.

Thank you for your help in this matter.

*Tim Mullen*
*Loveland, OH*

*Editor:*

The Tucson Sanyo Users Group is looking for new members. We meet the second Wednesday of the month at 7 P.M. Room 208, Geology Building, University of Arizona.

Persons wishing additional information may write: Julie Ingham, c/o Arizona Micro Exchange, 1927 E. Speedway, Tucson, AZ 85719 or call Bob Kalen (602) 742-0345 or myself (602) 721-9078.

*Gene McKee*
*Tucson, AZ*

*Editor:*

We have been selling the Sanyo MBC-550 series computers for one year now, and SOFT SECTOR for three months. We run a Bulletin Board during our off hours here at the store. I would like to let your readers know of our BBS and get the BBS added to your list. The hours are: 6 P.M. to 8 A.M. (Atlantic Standard) Our phone number is (902) 454-8344. Thank you.

*Basic Computer Sales*
*Halifax, NS*

---

# SOFT SOAPBOX

Success is a very subjective state of being. Everyone has his own idea about what success is and how to achieve it. Success can be achieved through financial gain, through critical acclaim or even, in some instances, through scandal.

I like to think that SOFT SECTOR has achieved its success through critical acclaim from you, the reader. It's heartening to get letters from people who say they heard about SOFT SECTOR through a user's group or they saw a friend's copy of the magazine.

SOFT SECTOR's continued success allows us to continue to bring you new services. The way we find out what services you want is through your "critical acclaim;" for example, "I love your magazine and it meets most of my needs, but I would like to see. . . ."

One way we are trying to meet the needs of our readers is through telecomputing. Telecomputing is yet another way to get maximum use from your computer. It's a way of sending and receiving information, sharing your own personal programming tips and meeting people — some whose face you'll probably never even see.

Are you wondering what really practical uses you can get from telecomputing? How many times do you get your copy of SOFT SECTOR and want one of the programs in it *immediately*? Do you get up and go directly to your computer to type it in? Probably not. Now you can have that program within a matter of minutes if you have the right equipment.

SOFT SECTOR programs are available through the CompuServe "Softex" service. You can download any of the programs found in the magazine if you have your own modem and terminal program. These programs are available for a nominal fee of $5 each. To some of you, this may seem high. However, if you want the program *now* instead of four hours later (when you finish typing it in) or a week after you receive your magazine (if you order it on SOFT SECTOR ON DISK), it will be well worth the price.

Telecomputing on the major information networks is not only convenient, but also fun. It's fun because there are unlimited possibilities. There are several Special Interest Groups (SIGS) which include collaborative novels, joke SIGs, and things such as forums, which allow you to go online and ask questions and the like. Information networks can help keep you informed and up-to-date about what's going on in the computer world. Along with the special interest groups are shopping services, bulletin boards and a world of other resources — and it's available 24 hours a day!

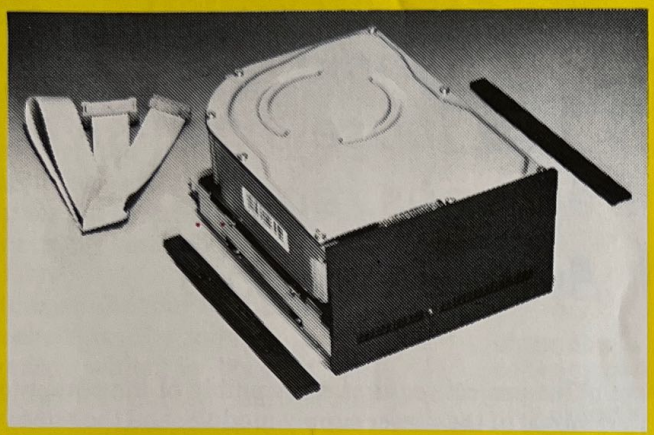
As we go along, SOFT SECTOR will become more involved in information network systems. For those who are interested and don't yet have the proper equipment, get moving! You will see several more services available through SOFT SECTOR and we don't want you to miss out on any of these. We think you'll be just as excited about what you see as we are and we hope you'll find it as useful as we intend it to be. For now, however, get your modem and your terminal program, look over what's available through the information networks and get ready to join us — online.

**Belinda Kirby**

# SANYO DEALERS —

## TO HELP IMPROVE YOUR COMPETITIVE EDGE — THESE FOUR BIG I's

**I**MPROVED DEALER PRICING/MARGINS — **I**MMEDIATE SHIPMENT
**I**MPROVED DEALER SUPPORT/CUSTOMER SERVICE — **I**MPROVED PROFITS



**MBC-775**
**Hard Disks 10-20MB**
Internal Shock Mounted
Low Power — Light Weight
External And Tape Backup
Available

**MBC-880**
**Hard Disks 10-120MB**
**NO** 32MB Limitation
30-85 MS Access Times
One Year Warranty
Tape Backup Available

**MBC-550/555**

**Memory Expansion**
Provides Up To 720K Usable Ram
Self-Contained For Quick/Easy
Customer Installation.
Power Supply And Ram Disk
Software Included.

**Hard Disk — Tape Backup**
10-40MB Formatted
Complete External System
File By File Or Streaming Backup
Combo Units Available

**Dealer** Support And Customer Service Direct From TW Technologies
**Dealer** Inventory And Associated Costs Reduced
**Dealer** Service Costs And Service Problems Reduced

Also Available At **VERY** Competitive Prices: HARD DISKS And
TAPE BACKUP For IBM And Most Other PC/AT Compatibles
Plus Hard Disks For Sanyo MBC-1000 Thru MBC-4000

## LEARN MORE!!!!
## FOR PRICES/TERMS CONTACT

**The SANYO Specialists**

TW TECHNOLOGIES, INC.
21636 North 14th Avenue, Suite A2
Phoenix, AZ 85027, (602) 581-0669

# Power Supply Upgrade

## *An inexpensive way to handle the power that those expensive add-ons require*

*By John Orlando III*

With the advent of the Video RAM Board and the popularity of 8087 based software like Turbo PASCAL, the Sanyo MBC-555 owner may find his machine "dead" after the addition of these power-hungry hardware accessories.

Specifically, a 555-2 with 256K RAM, a Video RAM Board, an 8087 co-processor and a serial board may not "boot" after the addition of the final possible board or the IC chip because the power supply will shut itself down! This is because the power supply sensed a larger than "normal" current draw, assumed it was a "short" and turned off the DC voltage to prevent damage. This could be the case, but is probably not in machines that have just had every possible option installed.

The "problem" is just that the power supply sensing circuit component values were selected, based on a guess of what the current draw of the future Video RAM Board would be. It is not that the power supply cannot supply the needed power, but that the current is being limited by the sensing circuit!

The "fix" to this problem is quite simple for those of us who can wield a soldering iron, and it will cost us a whopping 58 cents (plus tax) for the parts! In fact, because the parts are sold at Radio Shack in packages of two, we can do *two* machines for 58 cents!

---

*(John Orlando III is the director of engineering at Pocono Hospital in East Stroudsburg, Pa., as well as the owner of Effective Technologies, a consulting clinical engineering firm. He uses Sanyo computers at work, at home and in his business. John may be contacted at 1111 Mica Circle, Bethlehem, PA 18017; 215-691-2794.)*

This project requires disassembly of the computer, removal of the power supply module, and the removal or replacement of two resistors soldered to the PC board itself. If this is beyond your capabilities, show this article to a friend who happens to be in electronics or has de-soldered before! Also, if your computer is still under warranty, you had best not try this project and void the warranty. Finally, it is not necessary to perform this modification if you are not having any problems. It is only necessary if you expand your unit to its fullest capacity and find that it no longer operates. Some newer units, newer video boards or some memory/co-processor chips do not draw as much current, and will never require this upgrade. It is safer to leave this safety shutdown current as close to normal current draw as possible, so that whenever a problem arises, the computer shuts down automatically saving expensive components.

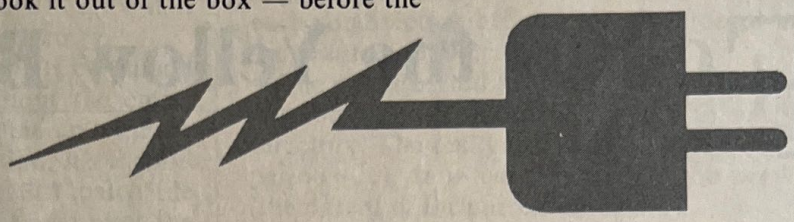
Before you begin, you must take a trip to your local electronics store or Radio Shack. Ideally, we need two ¼-watt five percent tolerance carbon resistors. One with a value of 1,000 ohms (1K ohm) and the other with a value of 91 ohms. Radio Shack sells #271-1321, which is a package of two 1,000 ohm ¼-watt resistors for 39 cents. The other resistor needed would ideally be 91 ohms, but this is an uncommon resistor from most shops and really not that critical. Radio Shack offers an 82 ohm resistor (#271-011) of ½ watt 10 percent tolerance. At 19 cents for a package of five, this component will suit our needs just fine.

Now it is time to warm up your soldering pencil and get out your Phillips-head screwdriver collection. As always, unplug your Sanyo from its AC power and all its peripherals. Now remove the cabinet cover

by removing the five Phillips-head screws as shown in the back of your user's manual.

1) Remove the two sets of cables (power and signal) that attach to the rear of the disk drives.
2) Remove the four Phillips-head screws that hold the floppy drive plate (on which both drives are mounted) from the main chassis. Note that the two screws on the power switch side also mount wire ties.
3) Remove the ground wire from the drive plate and slide the drives out toward the rear of the computer and place in a safe place.
4) Remove the four wire cables attached to the power supply board, which is between the power transformer and the fan on the bottom chassis.
5) Finally, remove the three Phillips-head screws that mount the power supply to the frame (two inside corners and center near the heat sink).
6) Remove the power supply and take it to your work bench or a friend's house.
7) Remove the power supply PC board from the metal frame of the power supply to expose the solder connections on the bottom of the board.
8) Locate resistor four (marked R4 on the component side which is the top of most boards). This should be a 2.4K resistor color-coded red/yellow/red/gold. Desolder it, remove and replace with your new 1K ohm resistor which should be color-coded brown/black/red/gold.
9) Next, locate resistor five (R5), which is a 68 ohm resistor color-coded blue/gray/black/gold. De-solder and replace with your new 91 or 82 ohm resistor.
10) Now solder all your connections. Clip leads as short as possible, and check to see that solder joints are neat and clean.

You are done! Now reverse the directions above and reassemble your computer. All connectors are polarized to go in only one way, so do not force. Be sure you do not pinch any wires when you reinstall the disk drive plate. Put the cover on, power up and your Sanyo should now boot like it did the first day you took it out of the box — before the additional 128K, before the serial board, before the 8087 and before the Video RAM board!



# Follow the Yellow Brick 'Path'

By Danny Humphress

In the September "Mastering MS-DOS," we began our explorations through a labyrinth of the MS-DOS hierarchal directory system. We continue our adventure this month as we discover special new commands designed to work with directories.

If you were with us on our most recent tour of MS-DOS, you will recall that MS-DOS version 2.11 allows you to create special files called *directories* which may contain other files and even other directories. The MS-DOS commands MKDIR and CHDIR allow you to create new directories and move from one to another.

**Killing Directories**

As discussed earlier, a directory is simply a file which contains other files. The similarity to a simple file ends here, though. When you use MS-DOS commands such as COPY and ERASE, the files *within* the directory are affected — not the directory itself. If you have a directory called AR and you enter the command ERASE AR, the *directory* would not be erased, but the *files* within that directory would be. Be careful. If you had other directories within AR, you would not be allowed to erase it at all until you removed those directories.

This brings us to a logical question: How do you remove directories? Not with ERASE! A special command, RMDIR (may be abbreviated RD), will remove a directory.

**Dead Directories Don't Have Files**

As a safety feature, RMDIR will only remove an empty directory. You must first use ERASE to remove all the files from the directory, and make sure that there are no directories below it. If there are other directories below the one you want to delete, you will have to go to the lowest directory and begin your destruction there. Once again, you may not remove a directory until it's completely empty.

Another RMDIR safety feature prevents you from ripping out the floor from beneath your feet. In other words, you cannot remove a directory that is your current default directory. To use RMDIR, you must move "up" at least one level (toward the root directory) from the directory you are removing.

Since you cannot go higher than the root directory, it is impossible to remove it. On the other hand, it *is* possible to remove the root directory using another method — erase the disk!

Now, let's remove our fictitious AR directory. We'll suppose that AR is a subdirectory of the root directory, that AR is our current default directory and that there are no directories within AR. These commands would get the job done:

```
CHDIR \          Move to the root directory
ERASE AR         Erase all files within AR
RMDIR AR         Remove the directory
```

**PATH**

They say, "build a better program and the world will beat a path to your directory."

Well, I may have taken a *few* liberties with that phrase, but only enough to serve my purpose. The problem with hierarchal directories, you see, is that one never seems to be in the same directory as the program you want to run. MS-DOS' "external" command programs such as CHKDSK, FORMAT and DISKCOPY are usually kept in the root directory. If you were in a lower directory and wanted to use one of these programs, you would normally have to move to their directory.

There *is* another way. The PATH command allows you to give MS-DOS a list of places to look when it can't find a program in the current directory. Once you've given it the list, it will remember it until you change it or reset the computer. In the above case, you would have wanted to tell MS-DOS to look in the root directory after searching the current directory.

For our above example, the PATH command would be:

```
PATH \
```

This tells MS-DOS to search the root directory (the backslash means "root") if it fails to find a program in the current directory.

As mentioned, you may specify an entire list of paths. MS-DOS will look in each one of them until it finds the

---

*(Danny Humphress is the owner of a computer software and consulting firm in Louisville, Ky. Danny brings to SOFT SECTOR his extensive experience with small business computers and applications software.)*

program you're trying to run. To do so, you just separate the paths with semicolons (;). Example:

```
PATH \;\UTILITY
```

The above command would have MS-DOS search the root directory and then a directory called UTILITY if it could not find the program in the current directory.

Remember, you only have to use the PATH command once. MS-DOS will remember it throughout the entire session with the computer. Once you've come up with the PATH sequence that best serves your situation, it's a good idea to make it a part of your AUTOEXEC.BAT batch file so that it is automatically set up each time you reset the computer (with that particular disk).

**Limitations**

You may have noticed that I used the words "command" and "program" a lot while talking about the PATH command. That is because they are the only types of files that PATH will work with. PATH will cause MS-DOS to search for "executable" (machine language) programs that have either a .COM or .EXE extension.

As a rule of thumb, unless you want to run a program that is executed by just typing the name at the MS-DOS prompt, PATH will not work for you. Likewise, most programs are not "smart" enough to use the path list you specified when they are looking for other programs and/or files. They will generally expect the files to be in the current directory.

MS-DOS external commands (those that are actually separate programs) such as CHKDSK, EDLIN, FORMAT, DISKCOPY, etc., all work well with PATH.

**Dot And Doubledot**

You may have noticed that when you did a DIR command on a subdirectory, there were two odd entries at the beginning of the list. The two directories '.' and '..' within each subdirectory are not *real* directories, but rather they are *pointers* to directories.

The single period (.) points to the current directory and may be used within any command to specify the *current* directory. The really useful "phantom entry" is the double period (..). It is used to specify the *previous* directory (the parent of the current directory).

Therefore, the command DIR . would give you a listing of the current directory, and DIR .. would give you a listing of the previous directory.

If you wanted to move up one directory but could not remember its name, CHDIR .. would take you there. Another CHDIR .. would move you up yet another level, and so on until you reach the root directory.

Once again, '.' and '..' may be used anywhere you would use full path names.

**Onward!**

With an understanding of hierarchal directory systems firmly seated in your mind, we are ready to explore deeper into MS-DOSdom. Next month, we'll search for the true meaning of device drivers. You know device drivers —those funny .SYS files that came on your MS-DOS disk. You'll know what they're for after you read the next "Mastering MS-DOS."

# Points, Lines And Boxes

## *Part III On Sanyo BASIC Graphics*

By Fred Blechman
Soft Sector Contributing Editor

As mentioned earlier, the Sanyo MBC-550 series microcomputers are always in the high-resolution graphics mode, capable of addressing and lighting any of 128,000 dots on the screen in any of the eight colors. You can also draw shapes made up of consecutive lines, draw rectangles by only specifying two corners, and then "fill" the rectangle with any color.

**"Setting" A Point**

The command used to identify a particular dot, or "point," is: PSET(X,Y),C which you can think of as "Point Set." By merely specifying the location in X and Y coordinates, and COLOR C, the point is lighted on the screen in that color. If you don't specify a color, then the current character color is used.

The range of values for the PSET horizontal X coordinate and the vertical Y coordinate is zero to 32,767 for each! However, anything beyond 639 for X or 199 for Y will fall off the screen unless an earlier WINDOW statement enlarges the screen dimensions to the values used for PSET. For all but advanced graphics, the WINDOW default value of zero to 639 for X and zero to 199 for Y is what you'll use, unless specified otherwise by a WINDOW statement.

Type and enter the following to PSET the screen corners in four different colors:

```
10 CLS
20 PSET(0,0),7
30 PSET(639,0),5
40 PSET(639,199),3
50 PSET(0,199),1
```

*(Fred Blechman, an acknowledged authority on home computers, has written several articles and books related to microcomputers. He is self-taught in electronics and computer BASIC programming and specializes in writing for beginners. Fred may be contacted at 7217 Bernadine Ave., Canoga Park, CA 91307; 818-346-7024.)*



If you are using a monochrome monitor, you may have to turn up the brightness to see the dot in the lower left corner.

Listing 1 draws eight sine waves on your screen, with short groups of different colored dots on a black background.

Lines 20 and 30 calculate the values for the coordinates, and Line 40 places the dot on the screen. The X MOD 7+1 at the end of Line 40 establishes the dot colors from one to seven, masking out zero.

**PRESET**

The "opposite" of PSET is PRESET, which plots a point using the background color. Since lighting a point with the background color makes it seem to disappear, this is effectively the "erase" command for dots that have been PSET, unless the background color has changed! If so, before PRESET use the COLOR statement with a comma and the desired background color number, such as:

```
COLOR,4:PRESET(X,Y)
```

This would change the background color to red without altering the foreground (printing) color, and then "erase" the point at X,Y that was on a red background. If you don't know the current background color, then use PRESET by itself, which will return the background color number, such as:

```
IF PRESET=5 THEN or PRINT PRESET
```

**Over And Then Down!**

Notice that, unlike the LOCATE command for specifying text position, where you go down and then over, the PSET graphics command goes over and then down!

At first this seems very inconsistent. It really makes a lot of sense, however, when you realize that locating text on a printed page is always specified first by line and then by character along that line. On the other hand,

mathematical graphing functions traditionally use the "Cartesian coordinate system" of specifying horizontal location and then vertical location.

### Inconsistencies

There are two inconsistencies you'll need to adjust to. One is that LOCATE commands start with one, and zero is not valid. Graphics commands, including PSET, start with zero.

Also, instead of the graphical "origin" (starting point) being at the lower left-hand corner of the screen, with vertical coordinates moving upward (common in math), the origin of the Sanyo 550 series (and many other computer screens) is the upper left-hand corner, with vertical movement downward. There are methods of translating coordinate systems to simulate real-world graphics, but they won't be covered here.

### Drawing LINEs

You can imagine the big task it would be to draw a line across the screen if you had to plot every point with PSET. Sanyo BASIC provides the LINE statement that will draw a line between any points within the world coordinate system (zero to 32,767 square). Remember, however, that an appropriate WINDOW must be specified or the line might be off your screen. Here, again, if you leave WINDOW and VIEW alone, you'll be operating within the normal zero to 639 and zero to 199 screen coordinate system.

The syntax for LINE offers various options:

```
LINE(X1,Y1)-(X2,Y2),C,B
```

The first coordinates, X1 and Y1, are the starting point. X2 and Y2 are the ending point. If X1 and Y1 are omitted, the ending coordinates of the last LINE or PSET (or 0,0 if no previous LINE or PSET) will be used as a starting point. The C, if used, specifies line color. The B, if used, means you want a box drawn using the starting and ending points as opposite corners. If you want the box filled with the designated color, add an F after the B.

Listing 2 uses LINE to create a box alternating in color from red to yellow.

Next month we'll continue our graphics lesson with circles, arcs and ellipses.

---

**Listing 1:** LIST-1.BAS

```
10 CLS
20 FOR X=0 TO 360 STEP .2
30 Y=50*SIN(X*8*3.14159/100)
40 PSET (X+100,Y+100),X MOD 7+1
50 NEXT X
```

**Listing 2:** LIST-2.BAS

```
5 CLS
10 LINE(100,100)-(50,50),4,BF
15 FOR D=1 TO 100:NEXT
20 LINE-(100,100),6,BF
30 FOR D=1 TO 100:NEXT
40 GOTO 10
```

# MichTron

576 S. Telegraph, Pontiac, MI 48053
Orders and Information (313) 334-5700
Prices Do Not Include Shipping and Handling

## SUPER DIRECTORY by Mark Feldman

Your disks are being overrun! Obscure files, chaotic paths, and disordered programs hide deep in mega-level subdirectories! It's *Super Directory* to the rescue; this sophisticated data-base will catalog your entire disk library!

- Catalogs 1.25 & 2.11 disks and hard-drives!
- Neat display shows full pathnames and lets you add disk number, category, and even remarks!
- Prints a variety of reports: from simple screen dumps to handy disk-sleeve sized print-outs.
- Sort, Find, and full-screen editing gives *Super Directory* the power to master the monsters of disorder, while mild-mannered menu-control keeps it friendly.

*Super Directory*, your ally against disk disorder!

128K Sanyo 550/555 or IBM compatible, MS-DOS 2.11 . . . . . $39.95

## DS-DOS PLUS 2.11 by Tim Purves

Don't settle for a DOS that's just "OK," boot-up with the best! Combined support of 80-track drives, 10 sector formats, extended memory, and IBM graphics is found only in DS-DOS PLUS 2.11.

- Controls any combination of drives: 40-track, s/d sided, 8, 9 & 10 sectors, AND 80-track, s/d sided, in 10 sectors.
- Handles up to 768K of memory extensions!
- Supports the SANYO VIDEO RAM BOARD. A simple DOS command toggles between Sanyo and IBM modes instantly! No re-booting!
- DS-DOS has the features that set the standards:

| | |
|---|---|
| DOS help file | Large cursor |
| Faster utilities | Remembers Date |
| Fast-printer patch | Directory sort |
| Menu-driven utilities | Text dump |

128K Sanyo 550/555, MS-DOS 2.11 . . . . $49.95

## CORNERMAN by J. Weaver Jr.

If a sloppy desk has you on the ropes, CORNERMAN is your ringside solution. It waits in memory while you use word processors, utilities, and even games! Then with a single keystroke, CORNERMAN brings you these great features:

- NOTEPAD lets you write notes for handy future reference and reminders.
- ASCII TABLE shows Sanyo symbols with their decimal & hex values. Great for programmers!
- PERPETUAL CALENDAR helps with planning to the year 2099.
- DIALER contains your personal phone book and can even dial for you!
- 8 DIGIT CALCULATOR: binary, octal, decimal, and hex; summation memory; math & logic functions; base conversions!
- Resumes previous program on command.

128K Sanyo 550/555 . . . . . $39.95

## EasyWindow by Paul McLeod

MichTron proudly announces the EasyWindow screen development system! EasyWindow lets C programmers design and implement user- interactive windows. End the hastle of creating friendly screens and input, and get back to designing your application!

- No more printf/scanf: EasyWindow makes INPUT/OUTPUT simple!
- Add text graphics and color to screen displays.
- Handles up to 64 windows at one time, with up to 255 I/O fields per window!
- Display windows virtually anywhere on the screen--even stacked or overlayed!
- Removes de-activated windows and updates underlying views automatically!
- Send windows to devices such as a printer.
- Use field types for data entry: text, decimal, etc.
- Special "groups" of related fields can even scroll independent of their window!
- Includes screen editor and library of C routines.

256K Sanyo 550/555 or IBM compatible, MS-DOS 2.11 and Lattice "C" 2.14 compiler. . . . $199.95

## EasyRecord by Paul McLeod

This binary-indexed record management system lets C programmers record and access file data quickly and orderly. EasyRecord manages up to 16 files at one time, maintains 8 key fields with 8 sub-fields each, accesses records with a variety of logic functions, and supports all C data types. Contains window-driven file setup program and complete library of C routines. Perfect for use with EasyWindow, or any data handling application.

128K Sanyo 550/555 or IBM compatible, MS-DOS 2.11 and Lattice "C" 2.14 compiler. . . . $199.95

## PICASSO by Bill Dunlevy

ONLY PICASSO IS 100% MACHINE LANGUAGE! It's the fastest, easiest, most powerful graphics program around! Pictures from PICASSO have even made the cover of Soft Sector--Twice! For marketing, design, or simple creative pleasure, PICASSO will knock your smocks off!

- Machine-language speed and power!
- PICTURE MENUS mean nothing to memorize!
- Rubber Band Graphics
- Two FULL-SIZED screens to draw on
- Pick from 64 colors, or make your own
- Choose brushes & densities for special effects
- SMARTFILL uses MULTICOLOR PARAMETERS!
- Draw ANY polygon: from triangles to ovals, arcs, and circles!
- Use stamps in your BASIC or ML programs!
- Use stamps in C & Pascal with Graphpac

256K Sanyo 550/555, Joystick recomm . . $99.95

***Why MichTron is number 1 is no ancient secret!***

# INPUT/OUTPUT

TIM PURVES
Soft Sector Contributing Editor

**Q.** *I just installed a Video RAM Board in my machine. When I try to load programs in BASIC, all I get is an error "Direct statement in file."*

*James Parlatore*
*Plainview, NY*

**A.** The problem sounds like you are trying to load Sanyo BASIC programs in GW-BASIC. The compression routines that are used by the two machines are quite different. The only way to load Sanyo BASIC programs into GW-BASIC is to first save them in ASCII using the command SAVE "FILENAME",A. Next load the program using GW-BASIC. Another way to get a "Direct statement in file" under some BASICs, is to have a line number zero.

---

*(Tim Purves is an expert on the Sanyo MBC-550/555 series computers and is an experienced programmer in assembly code, PASCAL, C and BASIC. He is available to answer any technical questions that the readers of SOFT SECTOR might have. All questions should be addressed to SOFT SECTOR, P.O. Box 385, Prospect, Ky. 40059.)*

**Q.** *I would like to buy a good assembler to use on the Sanyo. Which one would be a good buy?*

*Edward Solar*
*Dummer, NH*

**A.** I personally use Microsoft's *MASM* Ver 3.0. This is a very good assembler, and is reasonably fast and powerful. Another advantage is that it comes with a *Linker*, DEBUG and *Make* utility. This package requires DOS 2.11 or later.

---

**Q.** *When I press the NUM LOCK or GRAPH key in* WordStar *I get a strange set of characters (Vp6w6n). Why?*

*Vance Sheets*
*Convoy, OH*

**A.** This is caused by a bug in *WordStar*. See J. Weaver's excellent *WizStar* program and article for patching *WordStar's* "many features." (SOFT SECTOR, March '85, Page 8.)

---

**Q.** *Is the Keytronics 5150 DVORAK keyboard compatible with the Sanyo?*

*Bernard Tyrka*
*Watervliet, NY*

**A.** Neither the Keytronics 5150 nor the IBM-compatible will work with the Sanyo. This is due to the differences in the way the keyboard interfaces to the motherboard on both machines.

---

**Q.** *After installing the* Flight Simulator *patch, I cannot back up the disk on the Sanyo. What am I doing wrong?*

*Sidney Shore*
*Sharon, CT*

**A.** As I mentioned in the article, you cannot make copies of the disk on the Sanyo. This is because the Sanyo isn't fully IBM compatible. The only way to back up the *Flight Simulator* disk, is to find an IBM or fully compatible and make the backups there.

---

**Q.** *When asked how you found out the functions of Int 10h functions 70h to 73h, your answer was with* DEBUG. *Is there a command to interrogate* DEBUG *to get descriptions?*

*Ashley Campbell*
*Randolph, NH*

**A.** No, DEBUG knows nothing of the Sanyo and the special routines the Sanyo BIOS supports. The way to find out is to use the "unassemble command" to look at the code and analyze what the original programmer was trying to do.

---

**Q.** *What are the functions of the* FC.EXE *program on the DOS 2.11 disk?*

*Richard Downing*
*Australia*

**A.** FC.EXE is an intelligent file compare program. It has two modes; a binary mode and a text mode. In the binary mode, it compares the two files literally byte for byte. In the text mode, it has the option to ignore case and white space. Also, it will scan ahead and back to find a match. The format of the FC.EXE program is as follows:

```
FC [/#][/B][/W][/C] FILENAME1.EXT FILENAME2.EXT
/# How many lines to search forward and back for a match,
ranges from one to nine and the default is three.
/B Force a binary compare, all other options are ignored.
/W Ignore white space (tabs and spaces)
/C Ignore case (upper- and lowercase letters are considered
equal)
```

---

**Editor's Note:**
Due to space limitations, the *Volume* program promised last month will appear in the November issue.

# XREF

By William J. Locke

If you have never used a cross-reference utility before, get ready for a pleasant surprise. A cross-reference is one of the most useful programming tools you're likely to get your hands on.

What is a cross-reference? It is a list of variable names from a BASIC program. No big deal, right? Except each variable has next to it every line number from this program that the variable appears in. This utility also classes the variable into one of eight categories, simple - real, integer, double precision, and string, array - real, integer, double precision and string.

Have you ever wondered, when you're assigning a new variable name, if it has already been used somewhere in the program? This list will tell you. Have you ever needed to find every occurrence of a variable name in a program to implement a program change? This list will give you this information.

I think it wise to state from the beginning that this program is rather slow in execution. To cross-reference itself, this program takes about 10 minutes. The reasons for the execution time are twofold. First, it is written in interpreter BASIC and handles every character individually. A utility such as this, written in a compiled language would execute much faster. It is slow, but the price is right and the wait is worth it. Second, it is disk-based and this slows it down some more.

To use *XREF*, there is only one thing to watch for, and that is to make sure your type definition statements are at the very front of the program. If they aren't, the program may not classify your variables properly.

The first input is for output direction. The same thing applies for this program as for *Baslist* (SOFT SECTOR April '85, Page 60). Use PRN for the printer, AUX for the RS-232 or direct the output to a disk file.

The second input is the program to cross reference. This program must be saved using the ,A option so that it is in ASCII format. That's it! Prepare yourself for a really neat list of variables and associated line numbers.

The internal parts of the program are frequently commented on inside the program. I will go over the general flow of the program and point out some of the interesting parts.

Lines 10 through 150 control the main flow of the program. These lines call subroutines for a specific task as follows: Line 30 sets up the program. Line 50 reads a line from the file. Line 70 evaluates the line (this routine is the "meat" of the program). Line 90 tests for the end of the program file. Line 110 sorts the variables found in the program. Line 130 prints the results of the program. Line 150 ends the program.

The setup routine (lines 160 to 440) contains the definition, dimensioning and input statements. The program file is opened in Line 220. Lines 230 to 330 set up the sorted keyword list. The sorted keyword list is necessary to allow a timesaving keyword check. Lines 340 to 430 set up headings used in the final print.

Lines 470 to 480 read a program line into A$. This is the read line subroutine.

The next routine is the most complex. Its job is to take out the variables in the line and add the line number to a list when needed. In order to do this, we "weed" out words using a character-by-character comparison. This routine uses lines 490 to 1380.

The most interesting part of this routine is how we build our variable to test. Once this variable is built, we must test it to see if it is a keyword.

*(William Locke is self-taught on computers. He owns a business in Ocala, Fla. and does custom application software. He may be contacted at 14260 So. Magnolia Ave., Ocala, FL 32674; 904-753-4632.)*

# A Program Variable Cross-Reference Utility

CHAR$ holds the current character, XPOS holds the current position in the line, TEST$ holds the test string and A holds the ASCII value of the current character. Line 610 checks for double quotes; characters inside double quotes are ignored. Line 630 ends the routine if a single quote is encountered. Line 640 adds a legal character to the test string. Lines 650 to 660 handle the flow when a definition statement is in effect and call a routine (1030 to 1110) that sets the definition array. This array holds the definition value for every letter of the alphabet and the value is used later to type the variables.

Lines 670 to 740 reject certain unwanted test strings. These include numbers, Hex numbers, etc. Line 750 calls the routine that checks for key words. This routine (910 to 1020) deserves a little attention. RES holds the result of the routine, XCK the array number of the current check, XL the current number of the lowest check, XH the current number of the highest check and XCK1 the number of the prior check. This routine will check a list of strings (WORD$) for the occurrence of the test string (TEST$). The list must be in sort order. The comparison eliminates half the list at each check. This speeds the routine a lot. Also, the list may get very large and the time to check it will not get much larger.

The routine works as follows: Line 930 sets the result to not found. Lines 940 to 950 set the starting values for the rest of the variables. Line 960 checks to see if the same keyword is checked twice in a row, and ends the routine with the test string not found if this is the case. Line 970 sets the prior check equal to the current check. Line 980 sets the new check value. Line 990 tests for a greater than condition and if found sets the low number check and loops back to 960. Line 1000 tests for a less than condition and if found sets the high number check and loops back to 960. If the routine reaches Line 1010, then the test string is a keyword. This line sets the result to a found condition and Line 1020 returns to the calling routine.

This routine deserves a little more analysis than this. You may find use for this technique in other programs. XCK is the array element that we are comparing to the test string TEST$. Notice that the value of XCK is set halfway between XL and XH which are the upper and lower limits (these are set at the start of the routine to zero for the lower and the number of elements in the array plus one for the upper). On the first pass, XCK is set at half the array size. If the test string is greater than the array element, this tells us that the test string (if it exists in the array) is in the upper half. We then set XL=XCK and loop back. Then XCK is set halfway between XL and XH again and tested. This continues until the test string is found or the same array element is tested twice. If you actually need the array element number, XCK holds this value if the test string was found.

If you're paying attention, you will notice that each comparison eliminates half the remaining array. Comparing this method to a sequential check of a list containing 1024 elements, a sequential check takes an average of 512 comparisons to find a match. The average will actually be higher if you have a lot of cases where the test string is not found because it takes 1024 comparisons to check the entire list. This method takes at most 11 checks to find out that the test string is not in a list of 1024 — 10 at most if it is found. The really impressive concept is that if the array doubles in size, it will only take one more check to find the test string. The only problem is the list must be in sort order. Actually a "tree" method will work without sorting, but that's an article in itself. On with the program.

Upon return from the reserved word check, Line 760 checks to see if TEST$ is a variable. If it is, the routine to handle variables is called and then the program branches to Line 810. If the test string is a reserved word, Line 770 tests for a REM or DATA statement and ends the line check if found. Lines 780 to 800 check for a definition statement. Line 810 clears the test string and loops back to Line 560. Line 820 ends the line check routine.

The only other routine that deserves explanation is the variable handler (lines 1120 to 1380). This routine is called from Line 760 in the line check routine. Its job is to classify the variable and add it to the variable array. Lines 1140 to 1240 class the variable as integer, real, double precision or string. Line 1250 classes the variable as simple or array. Line 1260 either adds the line number to an existing array element or creates a new array element with the variable.

After the line evaluate routine comes the sort routine. You can write many articles on sort routines, however, this article is not one. This is simply a

# XREF

## A Program Variable Cross-Reference Utility

By William J. Locke

If you have never used a cross-reference utility before, get ready for a pleasant surprise. A cross-reference is one of the most useful programming tools you're likely to get your hands on.

What is a cross-reference? It is a list of variable names from a BASIC program. No big deal, right? Except each variable has next to it every line number from this program that the variable appears in. This utility also classes the variable into one of eight categories, simple - real, integer, double precision, and string, array - real, integer, double precision and string.

Have you ever wondered, when you're assigning a new variable name, if it has already been used somewhere in the program? This list will tell you. Have you ever needed to find every occurrence of a variable name in a program to implement a program change? This list will give you this information.

I think it wise to state from the beginning that this program is rather slow in execution. To cross-reference itself, this program takes about 10 minutes. The reasons for the execution time are twofold. First, it is written in interpreter BASIC and handles every character individually. A utility such as this, written in a compiled language would execute much faster. It is slow, but the price is right and the wait is worth it. Second, it is disk-based and this slows it down some more.

To use *XREF*, there is only one thing to watch for, and that is to make sure your type definition statements are at the very front of the program. If they aren't, the program may not classify your variables properly.

The first input is for output direction. The same thing applies for this program as for *Baslist* (SOFT SECTOR April '85, Page 60). Use PRN for the printer, AUX for the RS-232 or direct the output to a disk file.

The second input is the program to cross reference. This program must be saved using the ,A option so that it is in ASCII format. That's it! Prepare yourself for a really neat list of variables and associated line numbers.

The internal parts of the program are frequently commented on inside the program. I will go over the general flow of the program and point out some of the interesting parts.

Lines 10 through 150 control the main flow of the program. These lines call subroutines for a specific task as follows: Line 30 sets up the program. Line 50 reads a line from the file. Line 70 evaluates the line (this routine is the "meat" of the program). Line 90 tests for the end of the program file. Line 110 sorts the variables found in the program. Line 130 prints the results of the program. Line 150 ends the program.

The setup routine (lines 160 to 440) contains the definition, dimensioning and input statements. The program file is opened in Line 220. Lines 230 to 330 set up the sorted keyword list. The sorted keyword list is necessary to allow a timesaving keyword check. Lines 340 to 430 set up headings used in the final print.

Lines 470 to 480 read a program line into A$. This is the read line subroutine.

The next routine is the most complex. Its job is to take out the variables in the line and add the line number to a list when needed. In order to do this, we "weed" out words using a character-by-character comparison. This routine uses lines 490 to 1380.

The most interesting part of this routine is how we build our variable to test. Once this variable is built, we must test it to see if it is a keyword.

*(William Locke is self-taught on computers. He owns a business in Ocala, Fla. and does custom application software. He may be contacted at 14260 So. Magnolia Ave., Ocala, FL 32674; 904-753-4632.)*

modified shell sort which will serve our purpose.

Next is the print routine. Notice that we open a buffer for sequential output and use PRINT# statements instead of LPRINT statements. The output may go to a file or device.

The DATA statements are contained in lines 1800 to 1950. I think I have covered most any keyword you might use. If you see one pop up in your cross-reference, merely add it to the DATA statements here. Make sure the XOR keyword is last, or the program will not function properly.

## Variable Listing

**Simple Real Variables**

| | |
|---|---|
| A | ASCII value of CHAR$ |
| A1 | ASCII value used in definition handling |
| A2 | Detect value |
| Detect | Indicates variable type |
| I | Local counter |
| I1 | Counter in definition handler |
| Ignore | Toggle, one to ignore characters, zero to evaluate characters |
| IWord | Count of reserved words |
| J | Local counter |
| RES | Reserved word indicator |
| X1 | Number of elements in sort array |
| X2 | Sort increment |
| X3 | Length of variable name in variable handler routine |
| X4 | Length of line number in variable handler routine |
| XCK | Current array number in reserved word test |
| XCK1 | Previous array number in reserved word test |
| XH | Upper limit in reserved word test |
| XL | Lower limit in reserved word test |
| XLC | Line count in print routine |
| XLine | Line number in evaluate line routine |
| XPOS | Character position in line |
| XV | Array number in variable handler routine |

**Simple String Variables**

| | |
|---|---|
| A$ | Line contents |
| B$ | Line number |
| CHAR$ | Current character content |
| D$ | Local string manipulator |
| F$ | Output filename |
| F1$ | Input filename |
| M | Holding variable for sort routine |
| TEST$ | Test string for evaluate line routine |

**Array Real Variables**

| | |
|---|---|
| Detect | Definition values |

**Array String Variables**

| | |
|---|---|
| D$ | Holding array for definition handler |
| M | Variable names |
| MH | Headings |
| WORD$ | Reserved word list |

**The listing:** XREF.BAS

```
1Ø '
2Ø '*********************  CALL SETUP SUB  *************************
3Ø GOSUB 18Ø:CLS:PRINT "       STANDBY  -  WORKING":LOCATE 8,1Ø:PRINT "ON LINE NU
MBER:"
4Ø '*******************  READ LINE FROM FILE  *********************
5Ø GOSUB 47Ø
6Ø '*******************  EVALUATE LINE  ***************************
7Ø GOSUB 51Ø
8Ø '*******************  TEST FOR LAST LINE  **********************
9Ø IF EOF(1) THEN GOTO 11Ø ELSE GOTO 5Ø
1ØØ '*******************  SORT VARIABLES  *************************
11Ø CLS:PRINT "           VARIABLE SORT":GOSUB 141Ø
```

```
12Ø '******************  PRINT OUT RESULTS  ************************
13Ø GOSUB 157Ø
14Ø '******************  END PROGRAM  ******************************
15Ø CLOSE:END
16Ø '
17Ø '******************  SETUP SUBROUTINE  *************************
18Ø DEFSTR M
19Ø DIM M(3ØØ),WORD$(3ØØ),MH(13),DETECT(125,3)
2ØØ INPUT "ROUTE OUTPUT TO - PRN - X:FILENAME.EXT ";F$
21Ø INPUT "FILE TO CROSS REFERENCE ";F1$
22Ø OPEN"I",1,F1$
23Ø I=Ø
24Ø WHILE M(I)<>"XOR"
25Ø I=I+1
26Ø READ M(I)
27Ø WEND
28Ø IWORD=I
29Ø X1=I
3ØØ CLS:PRINT "              KEYWORD SORT":GOSUB 141Ø
31Ø FOR I=1 TO IWORD
32Ø WORD$(I)=M(I):M(I)=""
33Ø NEXT I
34Ø X1=Ø
35Ø M(Ø)="-1"
36Ø MH(Ø)="Simple Real Variables"
37Ø MH(1)="Simple Integer Variables"
38Ø MH(2)="Simple String Variables"
39Ø MH(3)="Simple Double Precision Variables"
4ØØ MH(1Ø)="Array Real Variables"
41Ø MH(11)="Array Integer Variables"
42Ø MH(12)="Array String Variables"
43Ø MH(13)="Array Double Precision Variables"
44Ø RETURN
45Ø '
46Ø '******************  READ LINE SUBROUTINE  *********************
47Ø LINE INPUT #1,A$
48Ø RETURN
49Ø '
5ØØ '******************  EVALUATE LINE ROUTINE  ********************
51Ø XLINE=VAL(A$):'                          Get line number
52Ø LOCATE 1Ø,15:PRINT XLINE
53Ø B$=STR$(XLINE)
54Ø IGNORE=Ø
55Ø XPOS=LEN(B$)-1:'                         Set Initial Position
56Ø XPOS=XPOS+1:'                            Increment Position
57Ø IF XPOS>LEN(A$)+1 THEN GOTO 82Ø
58Ø IF XPOS>LEN(A$) THEN A=58:GOTO 65Ø
59Ø CHAR$=MID$(A$,XPOS,1):'                  Get character from line
6ØØ A=ASC(CHAR$):'                           Set test value for character
61Ø IF A=34 THEN GOSUB 85Ø:GOTO 56Ø:'      Toggle Ignore if dbl quote
62Ø IF IGNORE=1 THEN GOTO 56Ø:'              Get next character if Ignore on
63Ø IF A=39 THEN GOTO 82Ø:'                  End routine if single quote
64Ø IF A=33 OR A=35 OR A=36 OR A=37 OR A=38 OR (A>47 AND A<58) OR (A>64 AND A<12
3) THEN GOSUB 89Ø:GOTO 56Ø:'                Add character to test var
65Ø IF A=58 AND DETECT>Ø THEN GOSUB 1Ø5Ø:DETECT=Ø:'      Reset Detect Var
66Ø IF DETECT>Ø THEN IF A=45 THEN GOSUB 89Ø:GOTO 56Ø:ELSE GOTO 56Ø:'
                                             Get next char if in def area
67Ø IF LEN(TEST$)=Ø THEN GOTO 56Ø:'          Ignore extra delimiter
68Ø IF LEFT$(TEST$,1)="&" THEN GOTO 81Ø:'    Ignore Hex Values
```

```
69Ø IF LEFT$(TEST$,1)="#" THEN GOTO 81Ø
7ØØ IF LEFT$(TEST$,1)="$" THEN GOTO 81Ø
71Ø IF LEFT$(TEST$,1)="%" THEN GOTO 81Ø
72Ø IF LEFT$(TEST$,1)="!" THEN GOTO 81Ø
73Ø IF LEFT$(TEST$,1)="Ø" THEN GOTO 81Ø
74Ø IF VAL(TEST$)>Ø THEN GOTO 81Ø:'          Ignore Numbers
75Ø GOSUB 93Ø:'                             Check for reserved word
76Ø IF RES=Ø THEN GOSUB 114Ø:GOTO 81Ø:'     If not then call variable handler
77Ø IF TEST$="REM" OR TEST$="DATA" THEN GOTO 82Ø:'End Routine if remark
78Ø IF TEST$="DEFINT" THEN DETECT=1:GOTO 81Ø
79Ø IF TEST$="DEFSTR" THEN DETECT=2:GOTO 81Ø
8ØØ IF TEST$="DEFDBL" THEN DETECT=3:GOTO 81Ø
81Ø TEST$="":GOTO 56Ø:'                     Clear test and loop to start
82Ø TEST$="":RETURN:'                        Exit routine - EVALUATE LINE
83Ø '
84Ø '*****************  TOGGLE IGNORE
85Ø IF IGNORE=Ø THEN IGNORE=1 ELSE IGNORE=Ø
86Ø RETURN
87Ø '
88Ø '*****************  ADD CHAR TO TEST
89Ø TEST$=TEST$+CHAR$
9ØØ RETURN
91Ø '
92Ø '*****************  TEST FOR RESERVED WORD
93Ø RES=Ø
94Ø XL=Ø:XCK=1
95Ø XH=IWORD+1:XCK1=Ø
96Ø IF XCK1=XCK THEN GOTO 1Ø2Ø
```

```
97Ø XCK1=XCK
98Ø XCK=INT((XH-XL)/2)+XL
99Ø IF TEST$>WORD$(XCK) THEN XL=XCK:GOTO 96Ø
1ØØØ IF TEST$<WORD$(XCK) THEN XH=XCK:GOTO 96Ø
1Ø1Ø RES=1
1Ø2Ø RETURN
1Ø3Ø '
1Ø4Ø '*****************  DEFINITION HANDLER
1Ø5Ø FOR I=1 TO LEN(TEST$)
1Ø6Ø D$(I)=MID$(TEST$,I,1)
1Ø7Ø A1=ASC(D$(I))
1Ø8Ø IF D$(I-1)="-" THEN FOR I1=A2 TO A1:DETECT(I1,DETECT)=1:NEXT I1
1Ø9Ø IF D$(I)<>"," AND D$(I)<>"-" THEN DETECT(A1,DETECT)=1:A2=A1
11ØØ NEXT I
111Ø TEST$="":RETURN
112Ø '
113Ø '********************  VARIABLE HANDLER
114Ø D$=LEFT$(TEST$,1)
115Ø A1=ASC(D$)
116Ø A2=Ø
117Ø FOR I=1 TO 3
118Ø IF DETECT(A1,I)>Ø THEN A2=I
119Ø NEXT I
12ØØ IF RIGHT$(TEST$,1)="%" THEN A2=1
121Ø IF RIGHT$(TEST$,1)="$" THEN A2=2
122Ø IF RIGHT$(TEST$,1)="#" THEN A2=3
123Ø D$=RIGHT$(STR$(A2),1)+TEST$
124Ø TEST$=D$
125Ø IF CHAR$="(" THEN TEST$="1"+D$ ELSE TEST$="Ø"+D$
126Ø I=1:XV=Ø
127Ø WHILE LEN(M(I))>Ø
128Ø X3=INSTR(M(I),",")-1
129Ø IF LEFT$(M(I),X3)=TEST$ THEN XV=I
13ØØ I=I+1
131Ø WEND
132Ø IF XV=Ø OR LEN(M(XV))>24Ø THEN GOSUB 135Ø:GOTO 134Ø
133Ø M(XV)=M(XV)+","+STR$(XLINE):IF XV>X1 THEN X1=XV
134Ø RETURN
135Ø X4=LEN(STR$(XLINE))
136Ø XV=I
137Ø M(XV)=TEST$+","+STRING$(7-X4," ")+STR$(XLINE)
138Ø RETURN
139Ø '
14ØØ '*********************  SORT ROUTINE  ********************************
141Ø LOCATE 1Ø,15:PRINT "PLEASE STANDBY - SORTING"
142Ø X2=2^INT(LOG(X1)/LOG(2))-1
143Ø LOCATE 5,6:PRINT "SORT INCREMENT IS   ";X2;"    NUMBER TO SORT IS  ";X1:LOC
ATE 7,6:PRINT "ON ELEMENT   ":FOR I=1 TO X1-X2:LOCATE 7,2Ø:PRINT "  ";I;"    "
144Ø IF M(I)<=M(I+X2) THEN GOTO 152Ø
145Ø M=M(I+X2):M(I+X2)=M(I)
146Ø IF I<=X2 THEN M(I)=M:GOTO 152Ø
147Ø FOR J=I-X2 TO 1 STEP -X2
148Ø IF M>= M(J) THEN GOTO 151Ø
149Ø M(J+X2)=M(J)
15ØØ NEXT J
151Ø M(J+X2)=M
152Ø NEXT I
153Ø X2=INT(X2/2):IF X2>Ø THEN GOTO 143Ø
154Ø PRINT "SORT COMPLETE":RETURN
```

```
155Ø '
156Ø '**********************  PRINT ROUTINE  **********************************
157Ø OPEN "O",3,F$
158Ø PRINT# 3,"                        VARIABLE LISTING FOR ";F1$:PRINT #3," ":XLC=XLC+2
159Ø FOR I=1 TO X1
16ØØ IF VAL(M(I))<>VAL(M(I-1)) THEN GOSUB 17ØØ
161Ø PRINT #3,MID$(M(I),3,79):XLC=XLC+1
162Ø IF LEN(M(I))>81 THEN PRINT #3,MID$(M(I),82,79):XLC=XLC+1
163Ø IF LEN(M(I))>16Ø THEN PRINT #3,MID$(M(I),161,79):XLC=XLC+1
164Ø IF LEN(M(I))>239 THEN PRINT #3,MID$(M(I),24Ø,15):XLC=XLC+1
165Ø PRINT #3," ":XLC=XLC+1
166Ø GOSUB 175Ø
167Ø NEXT I
168Ø GOSUB 178Ø
169Ø RETURN
17ØØ PRINT #3," ":PRINT #3," "
171Ø PRINT #3,MH(VAL(M(I)))
172Ø PRINT #3," "
173Ø XLC=XLC+4
174Ø RETURN
175Ø IF XLC<53 THEN RETURN
176Ø GOSUB 178Ø
177Ø RETURN
178Ø PRINT #3,CHR$(12):XLC=Ø
179Ø RETURN
18ØØ '
181Ø '**********************  DATA STATEMENTS  ****************************
182Ø DATA ABS,ALL,AND,AS,ASC,ATN,ATTR$,AUTO,BASE,BEEP,CALL,CDBL,CHAIN,CHR$,CINT,
```

```
CIRCLE,CLEAR,CLOSE,CLS,COLOR,COM,COMMON,CONT,COS,CREATE,CSNG,CSRLIN,CVD,CVI,CVS
183Ø DATA DATA,DATE$,DEF,DEFDBL,DEFINT,DEFSNG,DEFSTR,DELETE,DIM,EDIT,ELSE,END,EO
F,EQV,ERASE,ERL,ERR,ERROR,EXP,FIELD,FILES,FIX,FN,FOR,FRE,GCURSOR,GET,GO,GOSUB,GO
TO
184Ø DATA HEX$,IF,IMP,INIT%,INKEY$,INP,INPUT,INPUT#,INPUT%,INPUT$,INSTR,INT,KEY,
KILL,LEFT$,LEN,LET,LFILES,LINE,LIST,LLIST,LOAD,LOC,LOCATE,LOF,LOG,LPOS,LPRINT,LS
ET
185Ø DATA MERGE,MID$,MKD$,MKI$,MKS$,MOD,NAME,NEW,NEXT,NOT,OCT$,OFF,ON,OPEN,OPTIO
N,OR,OUT,PACK$,PAINT,PEEK,POINT,POKE,POS,PRESET,PRINT,PRINT#,PRINT%,PSET,PUT
186Ø DATA RANDOMIZE,READ,REM,RENUM,RESET,RESTORE,RESUME,RETURN,RIGHT$,RND,RSET,R
UN,SAVE,SEG,SGN,SIN,SPACE$,SPC,SQR,STEP,STICK,STOP,STR$,STRIG,STRING$,SUB,SWAP,S
YMBOL,SYSTEM
187Ø DATA TAB,TAN,THEN,TIME$,TINPUT,TROFF,TRONN,UNPACK$,USING,USR,VAL,VARPTR,VIE
W,WAIT,WEND,WHILE,WIDTH,WINDOW,WRITE,WRITE#
188Ø DATA STR$,PRINT#1,PRINT#2,PRINT#3,PRINT#4,PRINT#5,INPUT#1,INPUT#2,INPUT#3,I
NPUT#4,INPUT#5,WRITE#1,WRITE#2,WRITE#3,WRITE#4,WRITE#5
189Ø DATA PRINT#6,PRINT#7,PRINT#8,PRINT#9,PRINT#1Ø,INPUT#6,INPUT#7,INPUT#8,INPUT
#9,INPUT#1Ø,WRITE#6,WRITE#7,WRITE#8,WRITE#9,WRITE#1Ø
19ØØ DATA PRINT#11,PRINT#12,PRINT#13,PRINT#14,PRINT#15,INPUT#11,INPUT#12,INPUT#1
3,INPUT#14,INPUT#15,WRITE#11,WRITE#12,WRITE#13,WRITE#14,WRITE#15
191Ø DATA PRINT%1,INPUT%1
192Ø DATA CLOSE#1,CLOSE#2,CLOSE#3,CLOSE#4,CLOSE#5,PUT#1,PUT#2,PUT#3,PUT#4,PUT#5,
GET#1,GET#2,GET#3,GET#4,GET#5
193Ø DATA CLOSE#6,CLOSE#7,CLOSE#8,CLOSE#9,CLOSE#1Ø,PUT#6,PUT#7,PUT#8,PUT#9,PUT#1
Ø,GET#6,GET#7,GET#8,GET#9,GET#1Ø
194Ø DATA CLOSE#11,CLOSE#12,CLOSE#13,CLOSE#14,CLOSE#15,CLOSE#11,PUT#12,PUT#13,PU
T#14,PUT#15,GET#11,GET#12,GET#13,GET#14,GET#15
195Ø DATA TO,XOR
```

# CALL ADVENT

## SOFTWARE UTILITIES

**Uniform:** Your Sanyo 775 can read and write up to 80 CP/M disk formats.
**Uniform:** ..... **$69.95**

**Sidekick:** One of the most popular programs ever written. Use Sidekick as a calculator, notepad, appointment calendar, auto dialer, ASCII conversion table and much more. On-line help if you forget any of Sidekick's many functions.
**Sidekick (MS-DOS)** ..... **$54.95**

**SmartKey II+:** Saves you time. Makes every software program you use easier. Can reduce keystrokes by more than 50% by redefining any key to be any combination of characters or commands that you desire.
**SmartKey** ..... **$49.95**

**SmartPrint:** A powerful add-on to SmartKey, SmartPrint is a versatile writing tool designed to give you full access to your printer's features such as wide, bold, condensed, underlined, subscript, superscript, and more. Works great with programs like WordStar.
**SmartPrint** ..... **FREE WITH SMARTKEY PURCHASE**

**XTREE:** Directory maintenance program that graphically displays sub-directories and filename-paths. Complete control of your directory including delete, rename, view, list or show. A must for the 775.
**XTREE** ..... **$49.95**

## PROGRAMMING CORNER

**Toolworks Compiler:** This compiler is a complete subset of C. The two-pass compiler produces relocatable object files (.obj) which are compatible with the MS-DOS LINK program. Mathpak is included for true 32 bit floating point and signed intergers.
**Toolworks C Compiler - (MS-DOS)** ..... **$79.90**

**Turbo Pascal:** Borland version 3.0. The best Pascal compiler on the market. Period.
**Turbo Pascal (CP/M & MS-DOS)** ..... **$69.95**

**Turbo Toolbox:** Set of 3 utilities for use with Turbo Pascal.
**Turbo Toolbox (CP/M & MS-DOS)** ..... **$54.95**
**Turbo Tutor:** Teaches step-by-step how to use Turbo Pascal.
**Turbo Tutor (CP/M & MS-DOS)** ..... **$34.95**
**Turbo Graphics (MS-DOS)** ..... **$54.95**

## GAME ROOM

**Eliza:** The pioneering artificial intelligence program, Eliza is a psychiatrist that carries on a conversation in plain English about anything you wish. Great to play alone or in a group.
**Eliza (CP/M & MS-DOS)** ..... **$24.95**

**Adventure:** Explore the dangerous Colossal Cave for treasures. Endure mystical spells and overcome adversaries. The only original uncut version.
**Adventure (CP/M & MS-DOS)** ..... **$19.95**

**MyChess:** This award-winning world class chess program features nine skill levels with over 850 opening moves. Prints moves and saves game in progress.
**MyChess (CP/M & MS-DOS)** ..... **$34.95**

**Word Wiggle:** Scrambled letter game for building vocabulary skills of young and old. Eleven skill levels containing 40,000 word dictionary. Fast paced, fun, and educational. Great for kids!
**Word Wiggle (CP/M & MS-DOS)** ..... **$29.95**

## COOKS CORNER

**Computer Chef - #1 in the Series:** The computerized cookbook for food lovers. Contains a database of almost 100 recipes with space to add your own. Auto scale ingredient quantities, find recipes by key words or type of ingredients you want to use, then display or print your selection.
**Computer Chef (CP/M & MS-DOS)** ..... **$29.95**

**What's for Dinner - #2 in the Series:** 200 main dish and dessert recipes. Requires Computer Chef program (not included).
**What's for Dinner (CP/M & MS-DOS)** ..... **$19.95**

**The Best of Wok Talk - #3 in the Series:** Exotic Chinese cooking. Over 100 recipes from China. Includes Computer Chef program.
**The Best of Wok Talk (CP/M & MS-DOS)** ..... **$29.95**

**MyDiet:** Create your own personal nutritional profile that will help you derive the maximum benefit out of the foods you eat. MyDiet is designed for the professional nutritionist, but is easy to use. Compares to programs costing hundreds of dollars!
**MyDiet (MS-DOS)** ..... **$49.95**

## HOME & BUSINESS CENTER

**Checks & Balances version 3.6:** A complete personal checking program or business register. 3.6 is a complete package that includes check writer and data base. It is easy to learn and use with new features including more types of reports. An excellent money manager!
**Checks & Balances (CP/M & MS-DOS)** ..... **$74.95**

CP/M - DRI, MS-DOS - MicroSoft, Sanyo - Sanyo Business Systems Corp.

## WRITERS WORKSHOP

**Punctuation & Style:** Improves your writing by catching unbalanced quotes, parentheses and brackets, improper abbreviations, capitalization, sentence structure, much more. It's like having your own copy editor!
**Punctuation & Style (CP/M & MS-DOS)** ..... **$125.00**

**Word Finder:** This powerful 90,000 word Thesaurus allows you to select the best word for the application. Works inside WordStar for greater ease of use. Instantly searches its dictionary, then displays synonyms, and automatically deletes the "wrong" word and replaces it with the "right" word. Requires 380K disk storage.
**Word Finder (CP/M & MS-DOS)** ..... **$79.95**

**Wordpatch:** Print files with tiny, compressed, wide, or wide compressed type faces, 5 sizes of italic, real superscripts and subscripts, and 6, 7, and 8 lines per inch spacing. No new print controls to learn. Supports most popular dot matrix printers. A must for WordStar users!
**Wordpatch (CP/M & MS-DOS)** ..... **$49.95**

**The Word Plus:** The ultimate spelling checker. Not only finds misspelled words but also shows you correct spelling options, shows the word in context and much more.
**The Word Plus (MS-DOS)** ..... **$150.00**

**Spell:** Powerful spelling checker with over 50,000 words. Works great with Wordstar and other word processing programs. Faster than other spelling checkers usen on the Sanyo.
**Spell (CP/M & MS-DOS)** ..... **$49.95**

## SCHOOL HOUSE

**Reading Professor:** Now you can use your computer to boost your reading speed while increasing comprehension and retention! Self paced lessons show immediate improvement and feedback. Includes several different practice exercises. Developed by skilled educators, this program tracks user's progress and monitors reading rates up to 2500 wpm. Additional reading libraries available.
**Reading Professor (CP/M & MS-DOS)** ..... **$59.95**
**Additional Libraries** ..... **(each)** .... **$15.95**

**Math Tutors:** Improve your skills in basic thru advanced math. Learning thru multiple-choice problems with help and hint functions provided. Grades 8-College. (CP/M & MS-DOS)
**Algebra** ..... **$34.95**
**Calculus** ..... **$34.95**
**Geometry** ..... **$34.95**
**Trigonometry** ..... **$34.95**

**Funzeez:** Arithmetic and Language program to build basic skills, provides four different progams to choose from: flashing numbers, arithmetic drills, unscrambling words and word guessing. Grades 2-5.
**Funzeez (CP/M & MS-DOS)** ..... **$34.95**

**Call for other educational software.**

## HARDWARE & SUPPLIES

**Finger Print "Letter Writer":** Add near-letter-quality print. IBM and Apple Graphics printer emulation, plus 16 other print functions to your Epson FX series printer! Replacement ROMs, easy installation does not void printer warranty.
**Finger Print "LetterWriter"** ..... **$79.95**

**Diskettes, Double Density:**

| Maxell package of 10 with storage box. | | 3M box of 10 | |
|---|---|---|---|
| **Single Sided** | **$19.95** | **Single Sided** | **$22.95** |
| **Double Sided** | **$23.95** | **Double Sided** | **$26.95** |

**Economy Diskettes:** package of 25 includes tyvek sleeves.
**Single Sided** ..... **$29.50**
**Double Sided** ..... **$31.25**

**Call or Write for our FREE catalog**

All items are warranteed for 90 days. 30 day money back guarantee if not completely satisfied. Money back guarantee for software applies only if diskette seal is intact. VISA and Master Card are welcome. Please add $2.00 freight per total order and $2.00 for COD orders. California residents please add 6% sales tax. Prices, availability and specifications subject to change without notice.

**CALL TODAY**

**National** **(800) 821-8778**
**California** **(800) 521-7182**

**Hours: Mon - Fri 8 am - 5 pm PDT**

**DEALER INQUIRIES WELCOME.**

**advent products inc.**

3154-F E. La Palma Ave.
Anaheim, CA 92806
(714) 630-0446

# The Development Of Life On Your Sanyo

## *An enticing math game that has survived through generations*

By Jeff Sorensen and Phil MacKenzie

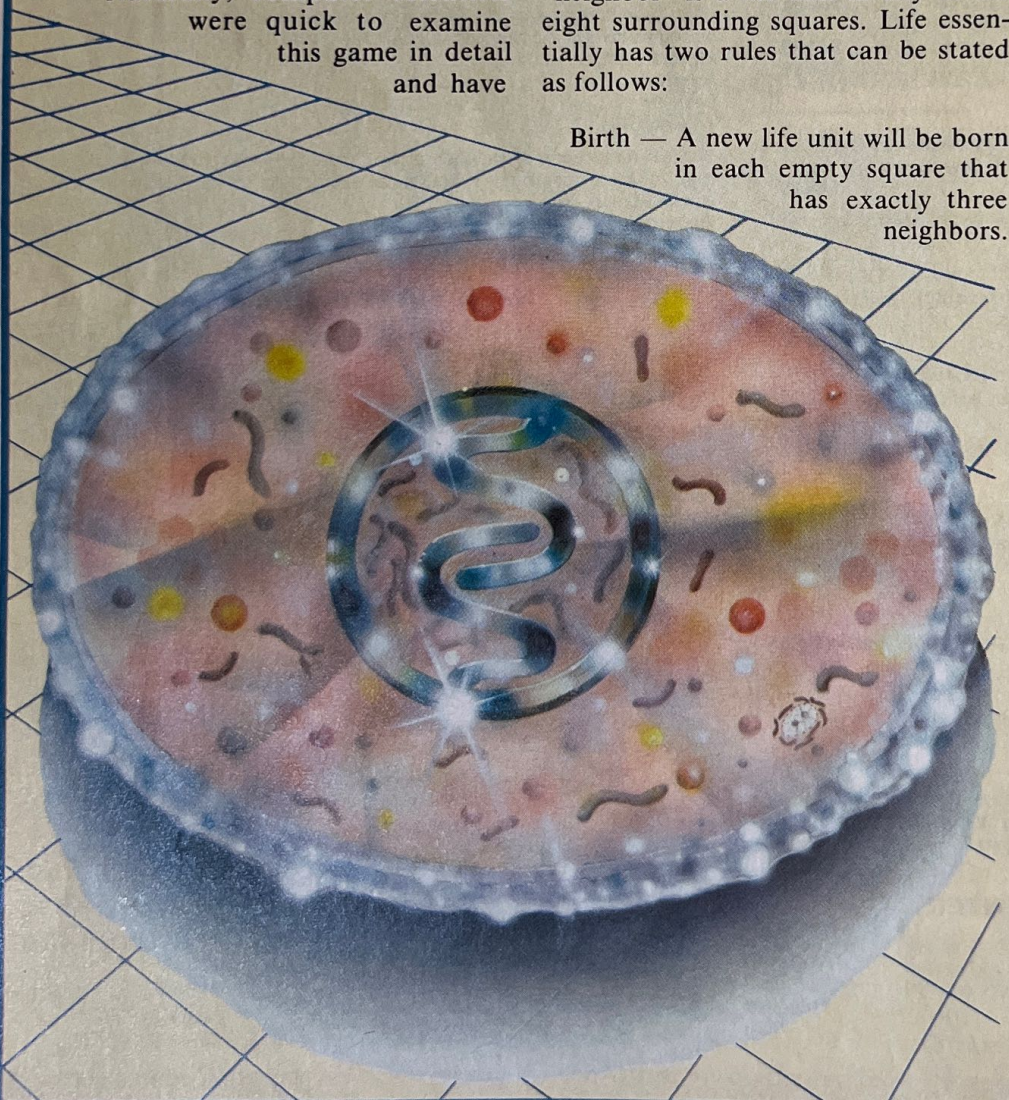
*Life* is a popular and peculiar mathematical game that was the brainstorm of mathematician John Horton Conway. It was originally practiced as a solitaire board game but is ideally suited to computer application due to its simple rules. Naturally, computer scientists were quick to examine this game in detail and have discovered countless fascinating properties, making *Life* far more than just a game.

Conceptually, *Life* is played on an infinite grid of squares (i.e., an infinite checker board). Each square can potentially contain a "life unit." A "neighbor" is a life unit in any of the eight surrounding squares. Life essentially has two rules that can be stated as follows:

Birth — A new life unit will be born in each empty square that has exactly three neighbors.

Death — An existing life-unit will die if it has fewer than two neighbors (isolation) or more than three neighbors (overcrowding).

All births and deaths occur simultaneously in what is called a "tick" or generation. Conway chose these rules to make the game both simple yet unpredictable. When played as a board game, executing these rules is both tedious and confusing; one mistake can ruin an experimental pattern. However, as a computer simulation, the complex patterns that develop are a joy to watch.



### Machine Language And BASIC

Obviously, *Life* on the Sanyo cannot be played on an infinite grid. For programming simplicity and memory conservation, the grid is limited to 78 by 48. However, the sacrifice in screen size is by far outweighed by the speed of each tick. The program can compute almost seven generations a second! In most *Life* simulations, the number of neighbors of each cell is calculated during each tick; this would be close to 30,000 calculations! However, this simulation maintains a table of existing neighbors and updates this with each birth and death, greatly reducing the number of calculations.

The BASIC program (Listing 1) will automatically load the machine language code into high memory. The machine language routines (Listing 2) are accessed by five different USR calls. The BASIC program provides convenient facilities to edit, save and study *Life* patterns. There is also a list of commands and their actions.

A life unit is represented by a red square. The cursor is represented by two vertical lines that appear to the left and right of a potential life unit. If testing a very large or complex pattern, it is advisable to save it. There is no way of backing up once a generation has passed, except to reload or redraw the pattern.

Once the machine language routines have been loaded by the first run, a RUN 300 command may be substituted for all subsequent runs. This will bypass

---

*(Philip MacKenzie and Jeff Sorensen are widely recognized in the Sanyo world. Philip is currently studying mathematics at the University of Michigan and Jeff is studying computer and systems engineering at Rensselaer Polytechnic Institute.)*

the routines to install the machine language program, thus speeding execution.

**Patterns Of Life**

The top line in Figure 1 shows a variety of common stable life patterns. The second line gives some examples of the more interesting, oscillating patterns. Many of these are common and will form from other patterns. The ones that oscillate over more than two ticks, however, are the most fascinating. These include the Pinwheel, Figure 8 and the Tumbler.

The third line illustrates several of the known patterns that replicate themselves as they move across the screen. These are some of the most fascinating of all patterns as their translational motion makes them seem almost alive. The Glider moves diagonally one square every four generations (one quarter the speed of light). The Spaceships and the Puffer Train move one square horizontally every two generations (one half the speed of light). The Puffer Train leaves behind a "cloud" of life units that eventually fade away, much like smoke from a train. A close observation of this form reveals two spaceships that play a crucial role in its motion.

Conway originally thought that all life patterns would eventually stabilize in a finite form. He was proven wrong with the discovery of the Glider Gun shown at the top of Figure 2. This remarkable arrangement shoots out a glider every thirty ticks!

The patterns in the bottom half of Figure 2 are two of the smaller Methuselahs. The Thunderbird has a life of 243 generations and the Bulldozer has a life of 148 generations. Both will fit on the Sanyo grid, though the Bulldozer must be placed in the lower right section of the screen.

**Life And Beyond**

Researchers have found that many complex patterns can assemble themselves from several carefully placed gliders. The gun shown can be generated by 13 strategically positioned gliders. As if anyone needed more gliders, a huge glider factory called a "breeder" was developed at Michigan Institute of Technology and is capable of generating thousands of glider guns!

Each generation of *Life* produces only one successor. However, some patterns can be created that have no predecessors, they must be directly created to exist. Hence, these patterns are named "Garden of Eden" patterns. They are extremely difficult to find. The only two that are known are 9 by 33 and 6 by 122. There are likely more such patterns, but to find them every previous possible generation must be examined — a lengthy procedure.

It is now known that on the *Life* grid, universal computers can be built using gliders as impulses. *Life* has also been used in work with self-replicating machinery. Models based on *Life* have been used in sociology, cosmology, automation and physics, to name just a few of the developments. One of the most impressive aspects of *Life* is the utter unpredictability of patterns that change under even simple rules. Certainly, the last page of the book of *Life* has not been written. There are still many unanswered questions.

This article was written to give you a "taste" of the mathematics which serve as the basis for the game of *Life*, but only by watching the program's hypnotic, almost organic, patterns can you appreciate the captivating nature of this thought provoking game.

## Command List

| | |
|---|---|
| Cursor (Arrow) Keys | Move cursor |
| | Return cursor to center of screen |
| + | Set a life unit at cursor position |
| - | Reset a life unit at cursor position |
| * | Erase the screen (requires confirmation) |
| 0 | Reset time to zero |
| ENTER | Start *Life* at full speed (will stop if stable or upon keypress) |
| . | Advance one *Life* generation |
| ? | Random screen generator* |
| H | Display a population histogram |
| G | Display grid limits |
| S | Save screen to disk |
| L | Load screen from disk |
| | Optional offset may be specified** |

*The density of the life units must be input as a decimal between zero (no units living) and one (all units living).

**The load offset parameter can be used to load patterns to different locations on the screen.

ex. Offset=80 Moves the pattern down one line
Offset=-15 Moves pattern left 15 positions
Offset=310 (80*2-10) moves pattern down two lines and left 10 positions

## Definitions

Garden of Eden pattern — a pattern which does not have a predecessor, and thus cannot be generated by any other pattern.

Generation (Tick) — the change in the grid pattern due to one simultaneous set of births and deaths.

Grid — ideally, the infinite checkerboard on which *Life* is played (on the Sanyo, it is a 78 by 48 checkerboard).

Life unit — a square on the grid which is alive. Represented by red squares on the Sanyo screen.

Life history — the complete set of generations from the original pattern to a final set of stable or oscillating patterns.

Methuselah — a pattern of less than 10 life units which does not stabilize in 50 generations.

Neighbor — a life unit in any of the eight surrounding squares on the grid.

Oscillating pattern — a pattern which repeats itself after two or more generations.

Population — the total number of life units on the grid.

Speed of Light — one square per tick, thus, the fastest speed at which any object can move.

Stability — the state in which all patterns on the grid are either stable or oscillating.

Stable pattern — a pattern in which no new life units are born and no existing units die.

For more information on this topic:

Berlekamp, Elwyn, John Conway, and Richard Guy. *Winning Ways*, Vol. 2. Academic Press, 1982.

*Byte*. Vol 3. December 1978.

Gardner, Martin. *Wheels, Life and Other Mathematical Amusements*. W. H. Freemen Company, 1983.

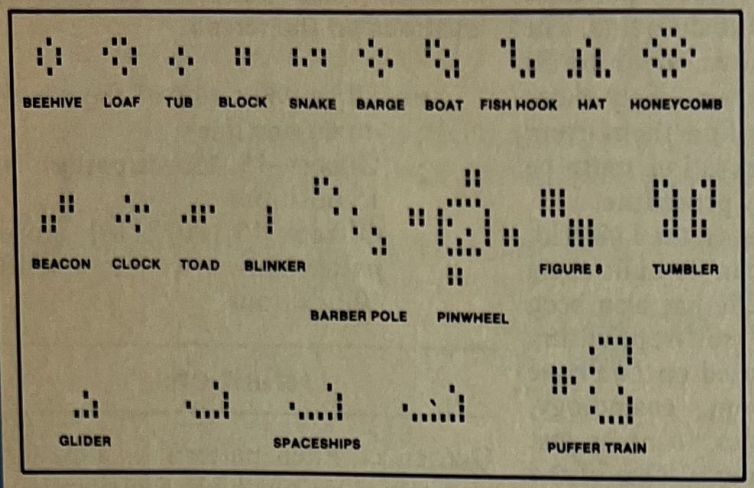



**Figure 1**

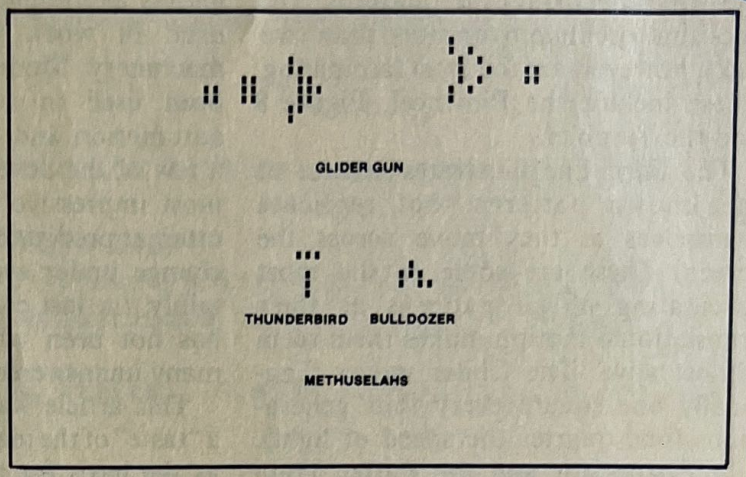



**Figure 2**

**Listing 1:** LIFE.BAS

```
1Ø CLEAR 581:DEF SEG=SEG(4):DEFINT A-X
2Ø CLS:LOCATE 12,18,Ø:PRINT "Please wait for machine language module to load."
3Ø FOR G=8ØØØ TO 828Ø:READ I:POKE G,I:NEXT G
4Ø FOR G=Ø TO 79:POKE G,32:POKE G+392Ø,32:NEXT G
5Ø FOR G=8Ø TO 384Ø STEP 8Ø:POKE G,32:FOR H=G+1 TO G+78:POKE H,Ø:NEXT H
6Ø POKE G+79,32:NEXT G
7Ø DATA 139,23,3Ø,14Ø,2ØØ,142,216,142,192,19Ø,Ø,Ø,191,16Ø,15,185
8Ø DATA 2Ø8,7,243,165,19Ø,8Ø,Ø,51,255,185,Ø,15,184,Ø,24Ø,142
9Ø DATA 192,138,132,16Ø,15,6Ø,3,116,49,168,16,116,84,6Ø,18,116
1ØØ DATA 8Ø,6Ø,19,116,76,254,76,175,254,76,176,254,76,177,254,76
11Ø DATA 255,254,76,1,254,76,79,254,76,8Ø,254,76,81,128,36,239
12Ø DATA 51,192,171,171,74,128,23Ø,127,235,42,254,68,175,254,68,176
13Ø DATA 254,68,177,254,68,255,254,68,1,254,68,79,254,68,8Ø,254
14Ø DATA 68,81,128,12,16,184,6Ø,6Ø,171,17Ø,71,66,128,23Ø,127,235
15Ø DATA 3,131,199,4,7Ø,226,154,31,137,23,2Ø3,139,7,3Ø,139,24Ø
16Ø DATA 2Ø9,224,2Ø9,224,139,248,131,198,8Ø,14Ø,2ØØ,142,216,184,Ø,24Ø
17Ø DATA 142,192,185,1,Ø,51,21Ø,246,4,48,116,174,235,217,139,7
18Ø DATA 3Ø,139,24Ø,2Ø9,224,2Ø9,224,139,248,131,198,8Ø,14Ø,2ØØ,142,216
19Ø DATA 184,Ø,24Ø,142,192,185,1,Ø,51,21Ø,246,4,16,116,184,233
2ØØ DATA 99,255,3Ø,14Ø,2ØØ,142,216,19Ø,Ø,Ø,185,16Ø,15,129,36,32
21Ø DATA 32,131,198,2,226,247,31,184,3,Ø,2Ø5,16,3Ø,14Ø,2ØØ,142
22Ø DATA 216,184,Ø,24Ø,142,192,51,255,139,215,19Ø,8Ø,Ø,185,Ø,15
23Ø DATA 246,4,16,116,11,38,199,5,6Ø,6Ø,38,198,69,2,6Ø,66
24Ø DATA 131,199,4,7Ø,226,234,233,11Ø,255
3ØØ DEFINT A-X:DEF SEG=SEG(4):DEF USRØ(A)=8ØØØ:DEF USR1(A)=8139
31Ø DEF USR2(A)=8174:DEF USR3(A)=8231:DEF USR4(A)=821Ø
32Ø X=39:Y=24:PT=USR3(Ø):DIM HS(127)
33Ø A$="##,###=t  #,###=Pop  &    &       Press ESC for help."+SPACE$(25)
1ØØØ HS(TM AND 63)=PT:KY$=INKEY$:IF KY$<>"" THEN 1Ø4Ø
```

```
1Ø1Ø C=6:GOSUB 211Ø:LOCATE 25,1,Ø:PRINT USING A$;TM,PT,V$;:V$=""
1Ø2Ø KY$=INKEY$:IF KY$="" THEN 1Ø2Ø
1Ø3Ø C=Ø:GOSUB 211Ø:GOTO 1Ø4Ø
1Ø4Ø KY=ASC(KY$):IF KY<12 THEN 1ØØØ
1Ø5Ø ON KY-11 GOTO 221Ø,231Ø:IF KY<27 THEN 1ØØØ
1Ø6Ø ON KY-26 GOTO 2ØØØ,222Ø,223Ø,224Ø,225Ø:IF KY<42 THEN 1ØØØ
1Ø7Ø ON KY-41 GOTO 241Ø,251Ø,1ØØØ,252Ø,261Ø,1ØØØ,271Ø,1ØØØ,1ØØØ,1ØØØ,222Ø,225Ø,2
23Ø,221Ø,224Ø
1Ø8Ø IF KY=63 THEN 281Ø ELSE IF KY<71 THEN 1ØØØ
1Ø9Ø IF KY>96 THEN KY=KY-32
11ØØ ON KY-7Ø GOTO 291Ø,3Ø1Ø,1ØØØ,1ØØØ,1ØØØ,311Ø:IF KY<83 THEN 1ØØØ
111Ø IF KY=83 THEN 321Ø ELSE 1ØØØ
2ØØØ REM *** HELP ROUTINE ***
2Ø1Ø CLS:RESTORE 2Ø4Ø
2Ø2Ø FOR G=1 TO 15:LOCATE 2+G,25,Ø:READ R$,S$:COLOR 4:PRINT R$;:COLOR 2:PRINT TA
B(4Ø)S$;:NEXT G
2Ø3Ø IF INKEY$="" THEN 2Ø3Ø ELSE PT=USR3(Ø):GOTO 1ØØØ
2Ø4Ø DATA Arrow Keys,Move Cursor,+,Set point,-,Reset point,*,Erase Screen,Ø,Rese
t time counter,,,".",Single Generation,RETURN,Continous Generations,,,?,Random s
creen generator
2Ø5Ø DATA H,Population history,G,Display grid limits,,,S,Save screen,L,Load scre
en
21ØØ REM *** CURSOR DRAW SUBROUTINE ***
211Ø LINE (X*8,Y*4)-(X*8,Y*4+2),C:LINE (X*8+7,Y*4)-(X*8+7,Y*4+2),C:RETURN
22ØØ REM   **** THE CURSOR MOVEMENT ROUTINE FOLLOW ****
221Ø X=39:Y=24:GOTO 1ØØØ
222Ø X=X-1:IF X<1 THEN X=78:GOTO 1ØØØ ELSE 1ØØØ
223Ø X=X+1:IF X>78 THEN X=1:GOTO 1ØØØ ELSE 1ØØØ
224Ø Y=Y-1:IF Y<Ø THEN Y=47:GOTO 1ØØØ ELSE 1ØØØ
225Ø Y=Y+1:IF Y>47 THEN Y=Ø:GOTO 1ØØØ ELSE 1ØØØ
23ØØ REM   **** THE CONTINOUS FRAME COMMAND (RET) ****
231Ø LOCATE 25,1,Ø:PRINT "Press any key to halt."SPACE$(54);
232Ø HG=USR(PT-32768!):IF HG<Ø THEN V$="Stable":GOTO 1ØØØ
233Ø PT=HG:TM=TM+1:HS(TM AND 127)=PT:IF INKEY$="" THEN 232Ø ELSE 1ØØØ
24ØØ REM   **** ERASE SCREEN COMMAND (*) ****
241Ø LOCATE 25,35,1:INPUT(1);"Erase screen? Confirm: ",Q$:IF Q$<>"Y" AND Q$<>"y"
 THEN 1ØØØ
242Ø PT=USR4(Ø):TM=Ø:GOTO 1ØØØ
25ØØ REM    *** SET/RESET COMMANDS FOLLOW (+,-) ***
251Ø PT=PT+USR1(X+Y*8Ø):GOTO 1ØØØ
252Ø PT=PT+(USR2(X+Y*8Ø)<>Ø):GOTO 1ØØØ
26ØØ REM    *** SINGLE GENERATION COMMAND (.) ***
261Ø HG=USR(PT-32768!):IF HG<Ø THEN V$="Stable" ELSE PT=HG:TM=TM+1
262Ø GOTO 1ØØØ
27ØØ REM   *** RESET TIME COMMAND (Ø) ***
271Ø TM=Ø:GOTO 1ØØØ
28ØØ REM   *** RANDOM SCREEN GENERATOR (?) ***
281Ø LOCATE 25,35,1:INPUT(5);"Random screen density (Ø-1): ",R$
282Ø Z=VAL(R$):IF Z<=Ø OR Z>1 THEN 1ØØØ
283Ø PT=USR4(Ø):FOR G=Ø TO 384Ø:IF RND(2)<Z THEN PT=PT+USR1(G)
284Ø NEXT G:TM=Ø:GOTO 1ØØØ
285Ø FOR I=MB TO TM:G=I AND 127
29ØØ REM   *** DISPLAY GRID LIMITS (G) ***
291Ø FOR G=12 TO 628 STEP 8:PSET (G,1),2:PSET (G,189),2:NEXT G
292Ø FOR G=5 TO 185 STEP 4:PSET (12,G),2:PSET (628,G),2:NEXT G:GOTO 1ØØØ
3ØØØ REM   *** POPULATION HISTORY (H) ***
3Ø1Ø CLS:MB=(TM>127)*(127-TM):MX=1:MN=4ØØØ:FOR G=MB TO TM
3Ø2Ø I=G AND 127:IF HS(I)>MX THEN MX=HS(I)
3Ø3Ø IF HS(I)<MN THEN MN=HS(I)
```

```
3Ø4Ø NEXT G:ZX=(MX-MN)/167:IF ZX=Ø THEN ZX=1
3Ø5Ø FOR G=MB TO TM:I=G AND 127:J=(G-MB)*4+32
3Ø6Ø LINE (J,167)-(J,167-(HS(I)-MN)/ZX),7,BF:NEXT G
3Ø7Ø FOR I=164 TO 4 STEP -1Ø:J=(164-I):SYMBOL (Ø,I),MID$(STR$(CINT(MN+ZX*J)),2),
1,1:NEXT I
3Ø8Ø FOR I=MB TO TM STEP 12:G=I-MB:LOCATE 23,INT(G/2)+3,Ø:PRINT I;:NEXT I
3Ø9Ø IF INKEY$="" THEN 3Ø9Ø ELSE PT=USR3(Ø):GOTO 1ØØØ
31ØØ REM   *** LOAD SCREEN COMMAND (L) ***
311Ø LOCATE 25,33,1:INPUT(1Ø);"Enter Load filespec: ",R$
312Ø IF R$="" THEN 1ØØØ ELSE INPUT(5);"  Offset: ",E$
313Ø I=VAL(E$):OPEN "I",1,R$
314Ø IF EOF(1) THEN 316Ø
315Ø INPUT #1,E$:G=VAL("&H"+E$)+I:IF G>384Ø OR G<Ø THEN 314Ø ELSE A=USR1(G):GOTO
 314Ø
316Ø CLOSE #1:PT=USR3(Ø):GOTO 1ØØØ
32ØØ REM   *** SAVE SCREEN COMMAND (S) ***
321Ø LOCATE 25,35,1:INPUT(15);"Enter Save filespec: ",R$
322Ø IF R$="" THEN 1ØØØ
323Ø C=Ø:OC$=",,,,,,,"+CHR$(13):OPEN "O",1,R$
324Ø FOR G=Ø TO 384Ø:A=USR2(G)
325Ø IF A THEN PRINT #1,HEX$(G)MID$(OC$,C+1,1);:C=C+1 AND 7:A=USR1(G)
326Ø NEXT G:PRINT #1:CLOSE #1:GOTO 1ØØØ
```

**Listing 2: Assembly Language Source Code**

```
                        ; *** NEWPOP=USRØ(OLDPOP) ***
                        ; ROUTINE TO ADVANCE SCREEN 1 GENERATION
1F4Ø  8B 17             USRØ:   MOV     DX,[BX]                 ;GET OLD POPULATION COUNT
1F42  1E                        PUSH    DS
1F43  8C C8                     MOV     AX,CS                   ;SET UP SEGMENT REGISTERS
1F45  8E D8                     MOV     DS,AX
1F47  8E CØ                     MOV     ES,AX
1F49  BE ØØØØ                   MOV     SI,ØØØØH                ;COPY NEWDATA TO OLDDATA
1F4C  BF ØFAØ                   MOV     DI,ØFAØH                ;FOR USE AS REFERENCE
1F4F  B9 Ø7DØ                   MOV     CX,2ØØØ
1F52  F3/ A5                    REP MOVSW
1F54  BE ØØ5Ø R                 MOV     SI,5ØH                  ;START WITH FIRST VALID
1F57  33 FF                     XOR     DI,DI                   ;ELEMENT
1F59  B9 ØFØØ                   MOV     CX,ØFAØH-ØAØH
1F5C  B8 FØØØ                   MOV     AX,ØFØØØH               ;SET ES TO POINT TO RED
1F5F  8E CØ                     MOV     ES,AX                   ;SCREEN
1F61  8A 84 ØFAØ        GEN1:  MOV     AL,[SI+ØFAØH]           ;GET OLDDATA VALUE
1F65  3C Ø3                     CMP     AL,3                    ;CHECK FOR BIRTH
1F67  74 31                     JE      BIRTH
1F69  A8 1Ø                     TEST    AL,1ØH                  ;CHECK IF LIFE UNIT
1F6B  74 54                     JE      GEN2                    ;PRESENT (JUMP IF NO)
1F6D  3C 12                     CMP     AL,12H                  ;CHECK IF SURVIVAL
1F6F  74 5Ø                     JE      GEN2                    ;IS SATISFIED
1F71  3C 13                     CMP     AL,13H
1F73  74 4C                     JE      GEN2
1F75  FE 4C AF           DEATH: DEC     BYTE PTR [SI-51H]       ;ADJUST ALL NEIGHBORING
1F78  FE 4C BØ                  DEC     BYTE PTR [SI-5ØH]       ;CELLS FOR DEATH
1F7B  FE 4C B1                  DEC     BYTE PTR [SI-4FH]
1F7E  FE 4C FF                  DEC     BYTE PTR [SI-1]
1F81  FE 4C Ø1                  DEC     BYTE PTR [SI+1]
1F84  FE 4C 4F                  DEC     BYTE PTR [SI+4FH]
1F87  FE 4C 5Ø                  DEC     BYTE PTR [SI+5ØH]
1F8A  FE 4C 51                  DEC     BYTE PTR [SI+51H]
1F8D  8Ø 24 EF                  AND     BYTE PTR [SI],ØFFH-1ØH  ;TURN OFF LIFE UNIT AND
1F9Ø  33 CØ                     XOR     AX,AX                   ;ERASE FROM SCREEN
1F92  AB                        STOSW
1F93  AB                        STOSW
1F94  4A                        DEC     DX                      ;ADJUST POPULATION COUNT
```

```
1F95  8Ø E6 7F          AND     DH,7FH                  ;ZERO STABLE FLAG
1F98  EB 2A             JMP     SHORT GEN3
1F9A  FE 44 AF  BIRTH:  INC     BYTE PTR [SI-51H]       ;ADJUST ALL NEIGHBORING
1F9D  FE 44 BØ          INC     BYTE PTR [SI-5ØH]       ;CELLS FOR BIRTH
1FAØ  FE 44 B1          INC     BYTE PTR [SI-4FH]
1FA3  FE 44 FF          INC     BYTE PTR [SI-1]
1FA6  FE 44 Ø1          INC     BYTE PTR [SI+1]
1FA9  FE 44 4F          INC     BYTE PTR [SI+4FH]
1FAC  FE 44 5Ø          INC     BYTE PTR [SI+5ØH]
1FAF  FE 44 51          INC     BYTE PTR [SI+51H]
1FB2  8Ø ØC 1Ø          OR      BYTE PTR [SI],1ØH       ;TURN ON LIFE UNIT
1FB5  B8 3C3C           MOV     AX,3C3CH                ;DISPLAY ON RED SCREEN
1FB8  AB                STOSW
1FB9  AA                STOSB
1FBA  47                INC     DI                      ;ADJUST POPULATION COUNT
1FBB  42                INC     DX                      ;ZERO STABLE FLAG
1FBC  8Ø E6 7F          AND     DH,7FH
1FBF  EB Ø3             JMP     SHORT GEN3
1FC1  83 C7 Ø4  GEN2:   ADD     DI,4                    ;LOOP FOR ENTIRE FIELD
1FC4  46        GEN3:   INC     SI
1FC5  E2 ØØ             LOOP    GEN1
1FC7  1F        RETBAS: POP     DS                      ;RETURN DX TO BASIC
1FC8  89 17             MOV     [BX],DX                 ;VIA USR ADDRESS
1FCA  CB                RET
                        ;
                        ;SET A PIXEL   USR1(X LOC + Y LOC *8Ø)    RETURNS 1 IF SUCCESFUL
                        ;
1FCB  8B Ø7     USR1:   MOV     AX,[BX]                 ;GET LOCATION OF PIXEL
1FCD  1E                PUSH    DS                      ;SAVE DS
1FCE  8B FØ             MOV     SI,AX
1FDØ  D1 EØ             SHL     AX,1                    ;FIND SCREEN ADDRESS
```

```
1FD2  D1 EØ                    SHL     AX,1                    ;4 * PIXEL ADDRESS
1FD4  8B F8                    MOV     DI,AX
1FD6  83 C6 5Ø                 ADD     SI,5ØH                  ;SET PARAMETERS FOR JUMP
1FD9  8C C8                    MOV     AX,CS                   ; TO BIRTH
1FDB  8E D8                    MOV     DS,AX                   ;RED SCREEN LOCATION
1FDD  B8 FØØØ                  MOV     AX,ØFØØØH
1FEØ  8E CØ                    MOV     ES,AX
1FE2  B9 ØØØ1                  MOV     CX,1
1FE5  33 D2                    XOR     DX,DX
1FE7  F6 Ø4 3Ø                 TEST    BYTE PTR [SI],3ØH       ;CHECK IF VALID POSITION
1FEA  74 AE                    JE      BIRTH                   ; YES
1FEC  EB D9                    JMP     SHORT RETBAS            ; NO, RETURN TO BASIC
                               ;
                               ;RESET A LIFE UNIT   USR2(X LOC + Y LOC *8Ø)   RETURNS 32767
1FEE  8B Ø7                    USR2:   MOV     AX,[BX]         ;GET SCREEN POSITION
1FFØ  1E                       PUSH    DS
1FF1  8B FØ                    MOV     SI,AX
1FF3  D1 EØ                    SHL     AX,1                    ;FIND SCREEN ADDRESS
1FF5  D1 EØ                    SHL     AX,1                    ;4 * PIXEL ADDRESS
1FF7  8B F8                    MOV     DI,AX
1FF9  83 C6 5Ø                 ADD     SI,5ØH
1FFC  8C C8                    MOV     AX,CS                   ;SET PARAMETERS FOR JUMP
1FFE  8E D8                    MOV     DS,AX                   ; TO DEATH
2ØØØ  B8 FØØØ                  MOV     AX,ØFØØØH               ;RED SCREEN LOCATION
2ØØ3  8E CØ                    MOV     ES,AX
2ØØ5  B9 ØØØ1                  MOV     CX,1
2ØØ8  33 D2                    XOR     DX,DX
2ØØA  F6 Ø4 1Ø                 TEST    BYTE PTR [SI],1ØH       ;CHECK IF VALID LIFE UNIT
2ØØD  74 B8                    JE      RETBAS                  ;NO, RETURN TO BASIC
2ØØF  E9 1F75                  JMP     DEATH                   ;YES
                               ;
                               ;CLEAR THE SCREEN AND REDRAW IT   USR4(Ø)
                               ;
2Ø12  1E                       USR4:   PUSH    DS
2Ø13  8C C8                    MOV     AX,CS                   ;ERASE ONLY BUFFER
2Ø15  8E D8                    MOV     DS,AX
2Ø17  BE ØØØØ                  MOV     SI,ØØØØH                ;OLDDATA LOCATION
2Ø1A  B9 ØFAØ                  MOV     CX,ØFAØH
2Ø1D  81 24 2Ø2Ø               ZEROL:  AND     WORD PTR [SI],2Ø2ØH     ;ERASE EVERYTHING BUT BORDERS
2Ø21  83 C6 Ø2                 ADD     SI,2
2Ø24  E2 F7                    LOOP    ZEROL                   ;ERASE WHOLE SCREEN BUFFER
2Ø26  1F                       POP     DS                      ;GO ON TO REDRAW ROUTINE
                               ;
                               ;REDRAW THE WHOLE SCREEN   USR3(Ø)   ->  RETURNS LIFE COUNT
                               ;     RED SCREEN MUST BE CLEARED PRIOR TO USE!!
2Ø27  B8 ØØØ3                  USR3:   MOV     AX,3            ;CLEAR SCREEN AND RESET IT
2Ø2A  CD 1Ø                    INT     1ØH                     ;TO ZERO POSITION IN ITS SEGMENT
2Ø2C  1E                       PUSH    DS
2Ø2D  8C C8                    MOV     AX,CS
2Ø2F  8E D8                    MOV     DS,AX
2Ø31  B8 FØØØ                  MOV     AX,ØFØØØH               ;RED SCREEN
2Ø34  8E CØ                    MOV     ES,AX
2Ø36  33 FF                    XOR     DI,DI                   ;DI=SCREEN POSITION
2Ø38  8B D7                    MOV     DX,DI                   ;DX=NUMBER OF LIVE UNITS
2Ø3A  BE ØØ5Ø R                MOV     SI,ØØ5ØH                ;START OF BUFFER
2Ø3D  B9 ØFØØ                  MOV     CX,ØFAØH-ØAØH           ;DON'T CHECK BORDERS
2Ø4Ø  F6 Ø4 1Ø                 DRAW1:  TEST    BYTE PTR [SI],1ØH       ;CHECK IF UNIT ON
2Ø43  74 ØB                    JE      DRAW2                   ;NO, SKIP
2Ø45  26: C7 Ø5 3C3C           MOV     WORD PTR ES:[DI],3C3CH  ;YES, DRAW
2Ø4A  26: C6 45 Ø2 3C          MOV     BYTE PTR ES:[DI+2],3CH
2Ø4F  42                       INC     DX
2Ø5Ø  83 C7 Ø4                 DRAW2:  ADD     DI,4
2Ø53  46                       INC     SI
2Ø54  E2 EA                    LOOP    DRAW1                   ;DO REST OF BUFFER
2Ø56  E9 1FC7 R                JMP     RETBAS
```

## TIME BANDIT Bill Dunlevy & Harry Lafnear

ACTION! Battle the Looking Lurker, Killer Smurphs, and others as you try to escape with the Treasures of Time. But hurry Bandit, the Technopatrol is hot on your trail!

ADVENTURE! Over TWENTY unique screens let you explore medieval dungeons, western frontiers, and future science in one game! And embedded somewhere within the 300 LEVELS OF TIME are the NINE GREAT ARTIFACTS! Can YOU find them all?

ART! This machine language epic will dazzle the fussiest graphic gawker and challenge the quickest arcade player. Beautiful scrolling landscapes, life-like animation, thrilling sound effects, and HUNDREDS OF DETAILED SCREENS -- The conquest of Time and Space awaits you...

128K Sanyo 550/555, Joystick recomm. . $39.95

## CHECKMATE

Chess players, prepare for the ultimate challenge! Announcing CHECKMATE! From the opening play you can see the computer "think" as it contemplates each move. This classic game has a level of play for every degree of skill: from novice, to tournament caliber for the expert! Superb graphics put you right across from the Master. And if you can't decide on a move, ask the computer to suggest one! You can even switch sides and save games to disk for future reference or replay. Match wits with the computer in the most challenging strategy game of all!

128K Sanyo 550/555 and MS-DOS 2.11 . . . . . $39.95

## SPEED RACER Bruce Hansen

The checkered flag drops as your pulse rises in this lively, new arcade game. The road twists to the horizon on a 3-D panorama that sets the stage for the most exciting race the Sanyo has ever seen! Vie for time as you speed through the curves. Stay ahead of the pack, but beware of those who will stop at nothing to slow you down! Four challenging raceways, complete with obstacles and 3-D scenery, put your skills to the test in this Turbo™ type game.

128K Sanyo 550/555, Joystick recomm. . $34.95

## THUNDER CHIEF McKenzie & Sorenson

Join the crew of the THUNDER CHIEF on a spectacular flight into danger, daring, and downright fun! Each mission in this 3-D jet simulator consists of another unique scenario: from defending American soil, to provocative attacks behind enemy lines--on Earth, and beyond! Can you overcome the challenge of swifter enemy missiles, tougher tanks, and smarter helicopters, or will you become a fireball over Moscow? Only your skill's the limit in this high-flying arcade game.

192K Sanyo 550/555, Joystick recomm. . $34.95

## PLEDGE by Jon Kring

Once the financial staff at your church tries PLEDGE, they'll vow never to go back to pen and paper. This new pledge-tracking program will take hours off recording donations, preparing reports, and mailing statements.

PLEDGE requires no computer knowledge and comes with friendly instructions. Data entry is a snap; easy-to-understand screens aid the user in typing data. Just a few keystrokes produce any of five neat, detailed reports: member rosters, pledge summary, collections, pledge vs. given, and year-to-date summary.

128K Sanyo 550/555, IBM compatible . $199.95

## TYPE RIGHT by Ken Olson

Whether you want to end the fumbling of hunt-and-peck typing, or simply improve your speed and accuracy, TYPE RIGHT is the answer. Learn touch-typing in a way that ends computer anxiety and keyboard fears. Written in MACHINE LANGUAGE to be fast and accurate, TYPE RIGHT never misses a key. Color graphics and large letters make TYPE RIGHT perfect for any age! Learn at your own pace; repeat a previous drill, or forge ahead to sample paragraphs. TYPE RIGHT offers friendly, menu-driven use, and keeps track of everyone's personal progress.

128K Sanyo 550/555 . . . . . $34.95

# MichTron

576 S. Telegraph, Pontiac, MI 48053
Orders and Information (313) 334-5700
Prices Do Not Include Shipping and Handling

*A utility that creates a menu of BASIC filenames from disk*

# BASIC Menu

By Ray C. Robinson

Wouldn't it be nice if a BASIC program could read all of the BASIC filenames directly from a disk, present them in a numbered menu for your selection and prompt patiently for your input? Think about it, no more manual updating of the menu when you add or delete files; the BASIC program would do all the work for you. Well, a BASIC program by itself can't do it, but one linked to a machine language program can, and that is exactly what MENU.BAS can do for you.

The heart of MENU.BAS is the USR function on Line 140. This function executes a machine language program which first searches through all the filenames on the disk, builds a buffer of filenames that have the extension .BAS and finally passes the list back to BASIC in the string variable A$. The number of BASIC filenames found is stored in memory and is passed to BASIC indirectly with the N=PEEK(3) statement. Listing 2 is the assembler source code that generated the object code used by the menu program.

The menu program is designed to work with MS-DOS version 2.11 and either Sanyo BASIC or GW-BASIC with the Video RAM Board.

Begin by typing in the BASIC program from Listing 1, using special care with the DATA statements on lines 350 through 410. These DATA statements make up the machine language code and an error here could spell disaster when the program is run. When you are sure the program is error free, save it under the name MENU.

If you have a Video RAM Board and want a version for GW-BASIC, save the program with the ASCII option (i.e., SAVE "MENU",A Then from DOS, put a copy of MENU.BAS on a disk that contains the video board system and GW-BASIC. Boot the system with this video board disk and from GW-BASIC, load the menu program and make the

*(Ray Robinson is an electronics technician with the FAA, holds a BA degree in business administration, and has been interested in home and personal computers since 1980. He may be contacted at Rt. 8, Box 725, Lake Charles, LA 70605; 318-474-5301.)*

**Listing 1:** MENU.BAS

```
1 '*************************************************
2 '*             BASIC MENU PROGRAM                *
3 '* BY RAY C. ROBINSON, LAKE CHARLES, LA. JULY 1985 *
4 '*************************************************
5 '
6 '
1Ø CLEAR 67
2Ø COLOR 6,Ø
3Ø CLS
4Ø M$="******BASIC MENU PROGRAM******"
5Ø PRINT TAB(2Ø) M$
6Ø PRINT:PRINT TAB(26) "ONE MOMENT PLEASE"
7Ø DEFINT I,H,V
8Ø RESTORE 35Ø
9Ø FOR I=Ø TO &H2:READ CF$:POKE I,VAL("&H"+CF$):NEXT
1ØØ FOR I=&HFC TO &H1Ø7:READ CF$:POKE I,VAL("&H"+CF$):NEXT
11Ø FOR I=&H1A1 TO &H1FD:READ CF$:POKE I,VAL("&H"+CF$):NEXT
12Ø DEF USR(A$)=Ø
13Ø A$=STRING$(248,32)
14Ø A$=USR(A$):N=PEEK(3):A$=LEFT$(A$,N*8)
15Ø DIM CH$(33)
16Ø CH$(1)="EXIT TO DOS":CH$(2)="EXIT TO BASIC"
17Ø FOR I=1 TO N :CH$(I+2)=MID$(A$,I*8-7,8):L=INSTR(CH$(I+2)," ")
18Ø IF L=Ø THEN 2ØØ
19Ø CH$(I+2)=LEFT$(CH$(I+2),L-1)
2ØØ NEXT
21Ø CLS:PRINT TAB(2Ø) M$
22Ø K=-11
23Ø FOR H=1 TO 41 STEP 2Ø
24Ø K=K+11
```

following changes to lines 10 and 20:

```
10 DEF SEG=4096
20 KEY OFF:SCREEN 0,0,0:COLOR
6,0,0:WIDTH 80
```

Change the fifth data statement in Line 360 from 02 to 01 so that it reads:

```
360 8B,DA,8B,5F,01 . . .
```

Save the corrected program without the ASCII option and your GW-BASIC version of MENU is ready to go.

The disk with the original Sanyo BASIC version of MENU is still in ASCII format, so boot the system with this disk and from Sanyo BASIC, load the MENU program and then save it without the ASCII option. Saved this way, the menu program will load faster and take up less disk space.

Now that all the hard work is done, put a copy of MENU.BAS on a disk that contains some of your favorite BASIC programs and run *Menu*. You can do this directly from DOS by typing BASIC"MENU for Sanyo BASIC or BASIC MENU (no quote) for GW-BASIC. When the MENU program runs, "BASIC Menu Program" and "One Moment Please" will appear on the screen, the disk will fire up, and the numbers one to 33 and BASIC filenames will be printed on the screen. The whole process takes two to five seconds, depending on how many BASIC filenames are found. To run the BASIC program of your choice, simply enter its menu number and press ENTER and the program will load and run automatically.

The first two menu choices will always be: "Return to DOS" and "Return to BASIC." With these choices available, all you need to do to your BASIC program listings is to change all END statements to RUN"MENU, then when your program ends it will return to the menu and you can either run another program or exit as you desire.

The maximum number of BASIC filenames the *Menu* program can handle is 31 and if your disk contains more names than this, they will just be ignored. You can open a slot in the menu by giving the MENU.BAS program a different file extension. For example, from DOS type: RENAME MENU.BAS MENU.MEN. Now the menu program will not appear in the menu but it isn't needed there anyway. If you do this though, you must use the full name MENU.MEN when you wish to run the program or replace the END statement in your program listings. This same technique could be used to hide any other basic filenames that do not need to appear on the menu.

## Program Explanation

| | |
|---|---|
| 10 to 110 | Clear memory and load machine language program at current DEF SEG. |
| 120 to 200 | Call machine language USR program and put BASIC filenames into array CH$( ). |
| 210 to 340 | Display menu of BASIC filenames, prompt for input and execute selection. |
| 350 to 410 | Data for machine language. |

```
25Ø FOR V=1 TO 11
26Ø LOCATE V+3,H,Ø:PRINT USING "## &                &";V+K;CH$(V+K)
27Ø NEXT V
28Ø NEXT H
29Ø LOCATE 2Ø,25,1:INPUT "YOUR CHOICE";CH$:C=VAL(CH$)
3ØØ IF C<1 OR C>N+2 THEN BEEP:LOCATE 2Ø,38:PRINT "    ":GOTO 29Ø
31Ø IF C=1 THEN 32Ø ELSE IF C=2 THEN 33Ø ELSE 34Ø
32Ø SYSTEM
33Ø END
34Ø RUN CH$(C)
35Ø DATA E9,9E,Ø1,ØØ,3F,3F,3F,3F,3F,3F,3F,42,41,53
36Ø DATA 8B,DA,8B,5F,Ø2,1E,53,Ø6,ØE,ØE,1F,Ø7,32,E4
37Ø DATA 2E,88,26,Ø3,ØØ,BA,21,Ø1,B4,1A,CD,21,BA,FC,ØØ,B4
38Ø DATA 11,CD,21,3C,FF,74,2B,BB,Ø4,ØØ,2E,FE,Ø6,Ø3,ØØ,BE
39Ø DATA 22,Ø1,8B,FB,FC,B9,Ø4,ØØ,F3,A5,83,C3,Ø8,BA,FC,ØØ
4ØØ DATA B4,12,CD,21,3C,FF,74,ØA,2E,AØ,Ø3,ØØ,3C,1F,74,Ø2
41Ø DATA EB,D8,Ø7,5F,BE,Ø4,ØØ,FC,B9,7C,ØØ,F3,A5,1F,CB
```

**Listing 2: Assembly Language Source Code**

```
BASDIR SEGMENT
MAIN PROC FAR
    ASSUME CS:BASDIR
START:    JMP   MENU
COUNTER   DB    Ø                    ;NO. OF .BAS FILENAMES
BUFFER2   DB    248 DUP(?)           ;.BAS FILENAME BUFFER
FCB       DB    Ø,"????????BAS"      ;FILE CONTROL BLOCK
          DB    25 DUP(?)
BUFFER1   DB    Ø                    ;DISK TRANSFER ADDRESS
PROGNAME  DB    127 DUP(?)

MENU:     MOV   BX,DX                ;BX NOW POINTS TO BASIC ARRAY
          MOV   BX,Ø2[BX]            ;LOCATION PASSED BY USR CALL
```

```
        PUSH DS                 ;
        PUSH BX                 ;SAVE CONTENTS OF DS,BX,&ES
        PUSH ES                 ;

        PUSH CS                 ;
        PUSH CS                 ;CS,DS,&ES NOW ALL SAME SEGMENT
        POP  DS                 ;
        POP  ES                 ;

        XOR  AH,AH              ;
        MOV  [COUNTER],AH       ;CLEAR COUNTER

        MOV  DX,OFFSET BUFFER1  ;FUNCTION 1AH
        MOV  AH,1AH             ;SET DISK TRANSFER ADDRESS
        INT  21H                ;

        MOV  DX,OFFSET FCB      ;FUNCTION 11H
        MOV  AH,11H             ;SEARCH FOR FIRST .BAS FILENAME
        INT  21H                ;

        CMP  AL,ØFFH            ;.BAS FILE FOUND?
        JE   DONE               ;NO-NO .BAS FILES ON DISK, EXIT

        MOV  BX,OFFSET BUFFER2  ;BX POINTS TO BUFFER2

AGAIN:  INC  BYTE PTR COUNTER   ;ADD ONE TO NAME COUNTER

        MOV  SI,OFFSET PROGNAME ;
        MOV  DI,BX              ;MOVE .BAS FILENAME TO BUFFER2
        CLD                     ;
        MOV  CX,4               ;
        REP  MOVSW              ;

        ADD  BX,8               ;BX POINTS TO NEXT FILENAME LOCATION
                                ;IN BUFFER2

        MOV  DX,OFFSET FCB      ;FUNCTION 12H
        MOV  AH,12H             ;SEARCH FOR NEXT .BAS FILENAME
        INT  21H                ;

        CMP  AL,ØFFH            ;.BAS FILENAME FOUND?
        JE   DONE               ;NO, EXIT PROGRAM
        MOV  AL,[COUNTER]       ;
        CMP  AL,31              ;31 FILENAMES YET?
        JE   DONE               ;YES, EXIT PROGRAM
        JMP  AGAIN              ;NO, FETCH ANOTHER NAME

DONE:   POP  ES                 ;
        POP  DI                 ;RESTORE REGISTERS AND MOVE
        MOV  SI,OFFSET BUFFER2  ;BUFFER2 TO BASICS STRING LOCATION
        CLD                     ;
        MOV  CX,124             ;
        REP  MOVSW              ;

        POP  DS                 ;RESTORE DS
        RET                     ;RETURN TO BASIC
MAIN ENDP
BASDIR ENDS
        END  START
```


Software review

# *Dear Word* — A Word Processor For A Sweet Price

*Dear Word* is an amazingly inexpensive yet capable word processor for the Sanyo MBC-550/555 and other IBM-compatible computers. It was tested on a Sanyo MBC-555 and a Citizen MSP-10 parallel printer.

I first wondered why anyone with a Sanyo and its bundled *WordStar* would ever want another word processing program. To my surprise, I found *Dear Word* working its way into many of my everyday working disks.

The program resides in only 27K of RAM. Quite an accomplishment for a full-featured word processor. This makes *Dear Word* easy to fit on a working disk that could use a word processor but is short on room. I use it as the resident word processor on my *Fancy Font* disk, for example. Honeybit Software wisely decided not to copy protect the program, so it can easily be tucked away on many working disks.

*Dear Word* is incredibly fast at everything. It is written in tight code which really makes it "fly." It allows renaming documents and deleting disk files from within a document — a lifesaver if you're out of disk space.

The program includes several features which are very handy for programmers. Files are easily converted back and forth from *Dear Word* to ASCII format. Entire paragraphs of text can be moved right and left for indentions such as subroutines in programs written in structured languages like *dBASE*.

The print options are somewhat limited, and not all control codes are available. Most of the features of my printer, which uses Epson codes, were supported.

The most amazing feature of the program is its price; only $15 by mail order from the publisher. The documentation comes as a 28-page *Dear Word* text file on disk that must be printed out.

**(Honeybit Software 5430 S.W. Helen Ave., Corvalis, OR 97333, 503-754-9124, $15)**

**— Stephen W. Baker**

Software review

# *Forbidden Temple* — An Ongoing Adventure

*Forbidden Temple* is a text Adventure. When the game opens, the first item on the agenda is to create "characters" — as many as desired. A "group" of between one and three is chosen, and the characters are given gold to buy weapons, armor, spells and potions. The group then enters the dungeon and moves around looking for monsters, which may be engaged in battle. Most of the monsters carry gold and a few carry special "orbs." When a group of monsters is killed, their goods are divided among the "good guys." The ultimate goal is to find all of the orbs, and to find a hidden door which will then unlock.

Graphics are displayed in a small "window" which shows the area of the dungeon just in front of the group of good guys. Most of the time, this contains a cryptic sketch of part of a hall which shows things like doors and corners. When monsters are engaged, the window shows a still picture of one of them — they are pretty ugly. The action is typical of text Adventures. When fighting, for example, each player is asked which monster he wants to aim for and the result is shown. The results of each monster's shot are then shown and the cycle repeats.

If the group manages to get out of the dungeon, the living members' gold is saved to disk and is available to those characters the next time the game is played. Also, when an orb is found, this fact is saved to disk. I like this feature of the game. At first, the characters do not have enough gold for more than one small weapon and no armor. As gold is accumulated, more effective items can be purchased, which changes the style of play. Progress is slow; it takes many hours to put a really effective team together.

*Forbidden Temple* is written in BASIC and run on a standard interpreter. This has advantages and disadvantages. The disadvantages are that the program requires 256K and loads slowly. It takes approximately four minutes for the interpreter to be called, the program loaded and the monster data files accessed. The advantage is that the source code is available, thus allowing the user to tinker with the program. For example, I didn't like the way some of the colors looked on my monochrome monitor, so I changed the COLOR statements. The availability of the source code and the interpreter also allows the user to cheat, but I won't go into that (my conscience bothers me too much).

I did not encounter any errors in the program (other than a few misspelled words), but there are a number of features I didn't like. There is some error trapping, but it is not complete. In setting up a group, for example, if the user enters the name of a character not yet created (or misspells a name), the program aborts, requiring the whole program to be restarted — this takes almost three minutes (unless the user knows enough BASIC to correct it in direct mode).

There are a few unnatural features. Sometimes, for instance, a character is asked if he wants to use a weapon or spend some gold not yet possessed. When a good guy

is killed, his gold is divided among the remaining members of the group. The gold the player had before that session, however, is still available on the disk for future sessions. Thus, the total holding of a group can be increased through the "death" of one of its members. Also, when expensive healing potions are purchased, the program deletes any potions the character already has.

The dungeon is very big, with lots of side passages and hallways. I found myself spending a lot of time wandering around looking for monsters.

Fighting is fun, but looking isn't. However, the size of the dungeon is a definite plus. It makes the game interesting — there are few things in life more frustrating than killing a whole bunch of monsters, getting lost in the maze and not being able to escape with the loot. The solution, I think, is to have more monsters.

Finally, the documentation is very good. The first page tells a story about how the world got to be the way the game says it is. It reads like the beginning of a continuing fantasy novel. The remaining few pages give a concise and clear account of the game and some advice.

**(Prickly-Pear Software, 2640 N. Conestoga Ave., Tucson, AZ 85749, 602-749-2864, $39.95)**

**— Stewart Shapiro**



Software review

# *SanyCad* — A Program For Play or Professionals?

*SanyCad* is a Sanyo-only computer-assisted drawing or drafting program which may well be suitable not just for hobbyists and those of artistic inclinations, but for more professional users as well. At least such is the outlook of Computer Associates in its release for this interesting software package, which is perhaps best described as a cross between a drawing or painting program, on the one hand, and a professional drafting program on the other. Indeed, after reviewing it, I still can't make up my mind as to which is the better description.

With *SanyCad*, as with other drawing programs, you can create lines, circles, arcs, boxes, paint in regions, and so on, either in color or shades (the latter for us poor cheapskates still using a monochrome monitor). The documentation helpfully repeats the instructions for resetting the infamous four DIP switches for proper use of color shades with a monochrome monitor. With a screen dump such as MichTron's *Freeze Frame* — the one cited in the documentation and which I happened to have a review copy of — you can make hard copy printouts of your creations. I found, as you will find with most screen dumps, there is some problem associated with circles and dimensions, but I don't think I can fault *SanyCad*. Moreover, Computer Associates explicitly suggests experimenting with a "printer ratio" so as to get an appropriate image from the screen onto paper.

The copy that I received consisted of two single-sided diskettes and a manual of about 30 pages. The first disk contains the main programs, and the second contained some small pictures which had been saved. Although small, these little images or overlays can be superimposed to create more impressive and involved drawings, particularly in conjunction with some powerful commands for moving and duplicating found on the drawing menu.

Using A-OK DOS 1 and single-sided drives, I was able to create an MS-DOS 1.25 system disk with BASIC and all the programs, as well as the overlays, as A-OK supports nine sector per track formatting. I mention this for three reasons: 1) You need not feel that two drives are needed. 2) You need not feel that you need a double-sided drive. 3) You can fit all the programs on a single disk with 180K. In any event, the transfer was easy enough (the most important and unusual thing was in copying *Freeze Frame*), and with the AUTOEXEC.BAT batch file, I had a self-booting system disk. With the menu structure of *SanyCad*, I just followed a few directions, and was then ready to begin.

First, I used the on-screen tutorial, which provided me with most of what I needed to know. True, I made a few mistakes, but I can't fault the software. The program responds as quickly as can be expected from a BASIC program involving graphics. The manual filled in some gaps and was adequate documentation-wise, although there are several typos or other grammatical errors which could use touching up. As for the rest of the package, there is

a warning about the licensing agreement and not opening the package if you cannot or will not respect the copyright and licensing. The software is not copy protected, although as it stands, the program is not listable. The manual does offer a friendly tone of willingness to help, and Computer Associates will call you back if it has to research a question for you.

One additional feature of *SanyCad* is a BASIC subroutine to allow you to use pictures you create and save in your BASIC programs. At review time, I did not have time to check this, but it seems like a nice extra. Of course, you can just SAVE and LOAD pictures using *SanyCad*, as I anticipate most people will do.

In actual use, you have access to a main menu of options, and a drawing submenu. One particular feature with which I was impressed is the ability to undo the last action. For instance, if you just added a line which you now decide you don't like, you can erase it. Sometimes, this erases part of something overlapping, but you can have *SanyCad* redraw the picture. You can magnify shapes, duplicate them and even move them. You can add text — courtesy of Sanyo BASIC's SYMBOL command. There is windowing capability and room for 24 screens in memory on a 256K machine. I like being able to use the arrow keys to move about the menus at the bottom of the screen, although the first letter of the command was often faster. I was sorry that a joystick is not supported. Other drawing programs, such as SoftSol's *Draw 'N Doodle*, allows toggling between the keyboard and a joystick. Still, it was not too inconvenient to use the keyboard, primarily the arrow keys to direct the crosshairs cursor.

One nice little extra, which I will have to try with other programs, is the use of the tab key (that's the key with the left and right arrows on it) to toggle between moving the cursor one pixel or eight pixels at a time. I was disappointed to see that, after virtually every action, it kept defaulting to one pixel, as this is very slow movement, and I'm an impatient sort.

There are other features which I have not touched on, such as the use of dimensioning (e.g. for drafting), but one can never cover everything in a small review like this. The most important question is whether you should buy it. This returns us to whom this is intended for. If you are only interested in some hobby play, a hundred bucks is a bit steep for the privilege. Honestly, it was fun. And true, *SanyCad* has some nice features not found in the less expensive offerings (such as the aforementioned *Draw 'N Doodle*), but I cannot say, with clear conscience, that they would justify spending more than perhaps $50. So, I would have to incline toward recommending this program for those who might use the program more professionally; some kind of art, drafting or the like. For them, the program will be helpful in their professional endeavors.

**(Computer Associates, Inc., Box 683, West Fargo, ND 58078, 701-281-0549, Sanyo MBC-550/555 with at least 192K, 256K recommended; $99.95)**

**— Dr. Michael W. Ecker**

Software review

# *Fancy Font* — A Touch Of Class

*Fancy Font*, a collection of programs and character sets developed by Soft Craft, Inc., provides you with the ability to get superb printing quality from a dot matrix printer. It also provides you with the ability to edit existing data sets or create new character data sets based on the particular requirements. When using this program you will find the output produced is fine letter quality resembling a typeset document, as opposed to a document printed on a dot matrix printer. Another capability provided is flexibility, which allows the user to create special characters required for company logos and technical documents which require unique symbols.

The package comes with existing character sets and works in conjunction with word processors, text editors and spreadsheet applications that produce standard ASCII files. This package is designed to work with the following printers: Epson FX and MX versions; IBM Graphics printer; Star Gemini-10X, 15X, Radix; Delta; NEC8023; and C.Itoh Prowriter. It runs under MS-DOS, requires 128K of memory and a minimum of 120K of disk storage.

Should you decide to purchase a copy of *Fancy Font*, you would receive two double-sided disks and the documentation. The disks contain the installation files for use if necessary. I did not require them. In addition to the installation files, there are help files, the executable programs *Pfont*, *Efont* and *Cfont*, a database of Hershey characters and various font files. The Hershey Character Database is a set of characters created by Alan V. Hershey for the National Bureau of Standards (Wolcott, NBS Special Publication No. 424). As stated in the documentation user manual, the font files provided are: Roman Subscript and Superscript 8 point; Roman 10, 12, 14, 18 and 24 points; Roman Italic and Bold 12 point; Old English 20 point; Script 20 point; and Special Characters 20 points.

In addition to the floppies, you will also receive one well-written user's manual, one quick reference card, a directory of files on each disk pack and two pamphlets. One defines the system requirements, a "getting started" tutorial, setting up requirements, examples and a guide to "trouble shooting." The other pamphlet is a library of additional utilities and fonts available for purchase.

The heart of *Fancy Font* consists of three programs *Pfont, Efont* and *Cfont*. *Pfont* is used for printing. It has three methods of operation; Interactive, Command Line and Parameter Input File. *Efont* is used to simply create new characters or to create new characters from existing fonts. *Cfont* allows the user to choose characters from the Hershey Database and map them to ASCII character codes to construct a font character set for use with the other *Fancy Font* programs.

*Pfont* is the primary program of the three. Basically, this program formats print text for a dot matrix printer. The input can come from a text file or interactively from the keyboard. Commands can be specified as a parameter for printing or they can be embedded in the body of the report. Some examples are:

1) `Pfont tty: +fo romn12` — This command instructs *Pfont* to use the typewriter feature where each line you type will be printed immediately using the Roman 12-point font.
2) `Pfont sample.ff +fo romni12` — This command instructs *Pfont* to print the file `sample.ff` using Roman Italics 12-point font.
3) To use embedded commands, first create a file on your favorite word processor like this:

```
\foThis line is an example \f1 of embedded
commands \f2 printing different fonts.
```

Save this file and then execute it using the print parameters specified as follows:

```
Pfont sample.ff +fo romn18 olde20 scrp20
```

*Pfont* is executed using `sample.ff` text file created by the word processor.

`\fo` told *Pfont* to print "This line is an example" using Roman 18-point font.
`\f1` told *Pfont* to print "of embedded commands" using Olde English 20-point font.
`\f2` told *Pfont* to print "printing different fonts" using Script 20-point font.

The Print Control Parameters provide the capability to use: font selection, rough draft, display text to screen as opposed to printer, print control parameters, page setup parameters, horizontal and vertical formatting and

parameters used for concatenating files and building strings of text. The Embedded Formatting Commands provide for font selection, format control, horizontal and vertical motion, character substitution, reverse background and comments.

*Efont* is used to create new characters or to edit existing fonts. *Efont* can be used to create company logos or other special graphics features. It can also be used to create foreign characters or accents. All *Efont* commands are single letters; they allow you to L-load a font, S-save a font, E-edit a set of characters, R-replace a set of characters, P-print characters and D-display characters on the CRT. T-test a font by printing it, Z-zap remove characters from the font, M-modify margins of a range of characters, V-specify a value to move characters up or down, A-display and modify font information and Q-quit.

The basic operating sequence is to: Create a text file for a character using the *Efont* edit commands, exit *Efont* and edit the character using a text editor or word processor and return to *Efont* and replace the original character with the modified characters.

*Cfont* is used to take existing characters from the Hershey database and map them to ASCII character codes to use with existing *Fancy Font* programs. The *Cfont* command interface is the same as *Efont*. All command names are single letters. Immediately upon entering the commands, the program prompts the user for any additional parameters required. The list of commands is: M-enter mappings from the Hershey Database, S-save the current mappings, L-load a set of mappings, Z-zap current mappings, G-generate a font with current mappings, F-set the scale factors for enlargement/reduction, B-set the baseline of the characters to be generated, P-print a range of Hershey characters and D-display the current mapping, scale factors and baselines.

The basic operation instructions for *Cfont* are: Execute *Cfont*, use the M command to begin mapping and *Cfont* then asks for the character the Hershey Character should be mapped to. This must be supplied in ASCII Code Decimal Equivalent. ASCII code conversions are found in Appendix eight of the user manual. *Cfont* then asks for the Hershey character. This must also be given in decimal equivalent. The complete Hershey Character Database is found in Appendix six of the user manual. The final step is to execute the G-generate command. This replaces the normal character with the Hershey character. Now when the font is used with *Pfont*, the Hershey character appears each time the assigned key of the character font is entered.

The user manual is well-written and contains all the information a user requires to understand *Fancy Font*. It contains explicit instructions for printing with fonts, editing fonts and creating fonts. The appendix consists of a glossary, font descriptions and samples, using fonts with word processors, error messages for each program, distribution file descriptions, Hershey database, data file formats, ASCII conversions and parameter and command summary.

*Fancy Font* is a powerful program that provides much flexibility for any user. However, because of its power and flexibility, it does require a learning curve. This does not mean the package is complicated or difficult to work with. However, it does mean that time will be required to master its many capabilities. The tutorial (supplied by the package) to get you started, only touches on the basics of this package, but it does get you thinking in the right direction. The user's manual explains each command and parameter of the three programs using the same format which is; the command or parameter, its usage, values used if any, examples and notes. The manual is consistent and complete and provides many examples. If an in-depth tutorial were supplied that tied the manual and the "getting started" tutorial together, this package would become easier to learn.

In summary, if you have a need for a powerful package that can produce output on your dot matrix printer that resembles typeset and have a need to type technical papers requiring special characters, then you should consider this package. It does work well and it becomes easy to use with some practice.

**(Soft Craft, Inc., 222 State Street, Suite 400, Madison, WI 53703, 608-257-3300, $180)**

**— John E. Schlosser**

## Round and round we go to get . . .

# High Resolution Daisy Wheel Graphics

*By Mick and Jon McGuire*

If you have a daisy wheel printer that allows variable horizontal and vertical spacing, you can dump your screen to your printer and get very nice graphics and enlarged text on paper.

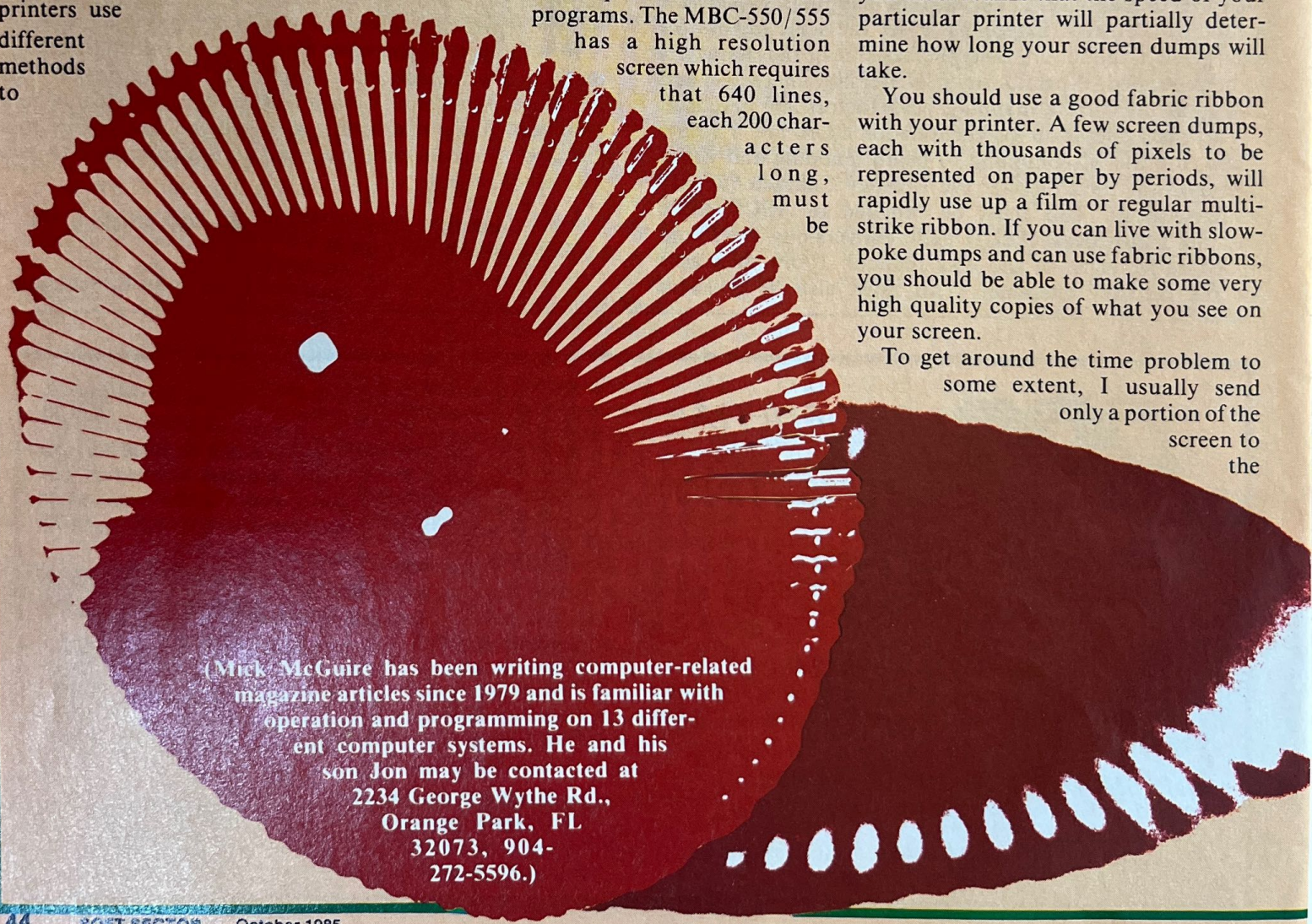
Graphs, charts and just about anything else on screen can be dumped to your daisy wheel printer *if* your printer has a programmable Horizontal Motion Index (HMI) and Vertical Motion Index (VMI). Not all daisy wheel printers can do this, but most of the newer ones have this feature. My Silver-Reed EXP-400 printer, which was very inexpensive, has variable HMI and VMI. Different printers use different methods to set HMI and VMI, so you may want to modify the listings with this article to accommodate your particular printer. Consult your printer manual for details.

That was the good news. The bad news is that the process is very slow! It takes a long time to scan and decode the screen, and it takes the daisy wheel printer a long time to print all those dots necessary for the graphics. In the graph and star figures used as illustrations (Figures 1 and 2), the time necessary to dump the screen to the printer was almost equally divided by the scan and print sections of the programs. The MBC-550/555 has a high resolution screen which requires that 640 lines, each 200 characters long, must be scanned to dump a full screen. Given the slow speed of a BASIC scan and the slow print speed, you can see why it takes a long time.

*Theoretically*, at a print speed of only 10 CPS, a full screen printout should take three and one-half hours, but in fact, it requires a lot less time due to the fact that my printer does not pause to "print" spaces, only formed characters (periods in this case). A really dense graphics screen printout will take a very long time, but most of my full screen prints average 15 to 45 minutes per screen. It's not as bad as it sounds, but you must realize that the speed of your particular printer will partially determine how long your screen dumps will take.

You should use a good fabric ribbon with your printer. A few screen dumps, each with thousands of pixels to be represented on paper by periods, will rapidly use up a film or regular multistrike ribbon. If you can live with slowpoke dumps and can use fabric ribbons, you should be able to make some very high quality copies of what you see on your screen.

To get around the time problem to some extent, I usually send only a portion of the screen to the



(Mick McGuire has been writing computer-related magazine articles since 1979 and is familiar with operation and programming on 13 different computer systems. He and his son Jon may be contacted at 2234 George Wythe Rd., Orange Park, FL 32073, 904-272-5596.)

printer. Time is not wasted scanning and printing "empty space." An entire screen *can* be printed, but this is usually not needed or desired. The sample illustrations (Figures 1 and 2) took seven and 22 minutes to print. If you plan on doing a lot of screen dumps or are the impatient type, save your dimes, quarters and dollars and buy a high speed dot matrix or ink jet printer. On the other hand, if you need high quality screen dumps only "once in a blue moon," use your daisy wheel and save that money for something else.

You can cut the required time almost in half if you use a machine language scan of the screen, but I did not include one as I wanted the programs presented here to be understandable and easily modifiable for other computers as well as the Sanyo series computers.

**Physical Constraints**

One thing to consider before starting: Will a full screen dump fit on a standard sheet of paper? How wide is your printer platten? With my printer, which has a short platten and can only print eight-inch lines, a full screen dump from my Sanyo MBC-550 will not fit. As I stated earlier, the Sanyo has a screen which is 640 pixels wide. Through trial and error, I learned that I needed a printer capable of printing a 13.4-inch line, or I had to figure a way to print the screen sideways. Since the screen is only 200 pixels high, I decided to scan the Y axis instead of the X for each printed line. The only drawback to this is that I lost some flexibility as there is less control of the VMI, since it is variable in increments of $^1/_{48}$ inch as opposed to $^1/_{120}$ inch for the HMI. There is a positive side effect in that the vertical scan of 200 pixels is easily converted into a string whose length is under the maximum allowed for a "normal" string variable (255). That means that each scan can be represented by a single variable instead of the three it would take to store a horizontal scan.

**Program Listing 1**

Listing 1 is a sample program which will draw and dump the star shape found in Figure 1. Lines 10 through 135 will draw the star, and lines 10000 through 10110 will scan the screen area that the star occupies and send it to the printer. Lines 5 and 145 will keep track of how long the whole process takes. The HMI and VMI controls, found in lines 10010 and 10020, are for a Silver-Reed EXP-400 daisy wheel printer, and might have to be changed to reflect your particular printer requirements. The REM statements in the other listings explain what most of the lines in the subroutine (10000 through 10110) do, so I won't elaborate on this listing any further.

**Program Listing 2**

For a Sanyo MBC-550/555 (or other computers with a BASIC GCURSOR command), you can use this program to define an on-screen "box" which will contain the graphics or text that you wish to scan and send to the printer. This is very handy, as the GCURSOR lines do not interfere with what is on the screen, but do allow you to select the area to print. Use this listing as a subroutine. Your main program will have to contain a GOSUB 63000 command.

Once you are operating in the subroutine, two lines will appear. Both lines can be controlled by pressing the arrow keys on the numeric keypad. The cursor lines normally increment by one pixel at a time, but this can be changed to 10-pixel jumps if you press the TAB key once. Pressing the TAB key again returns you to the one-pixel mode.

Position the GCURSOR lines so that they intersect at the upper left corner of the "box" you want to define. The lines will show you where the left and top sides of the box are. Press the ENTER key once to store this position, then use the arrow keys to move the line intersection to the point on the screen that will correspond to the bottom right corner of your "box." Press ENTER once again and the program will scan and print what is inside the boundaries you have defined. After printing is completed, the RETURN in Line 63150 will send you back to the main program. The zero (0) in Line 63100 is the color code for black. This should be changed to match the color code of whatever you use for background color on the screen. The Sanyo MBC-550/555 computers use black (0) as the background color default.

As with the other listings, do not type in the REM lines. All important lines are incremented by 10 so that the AUTO line numbering function can be used.

Caution! Screen dumps using your daisy wheel printer *may* prematurely wear out your print wheel. While I have been using my Silver-Reed EXP-400 for several months for screen dumps, I am not sure how many times each leaf of the wheel can be hit by the hammer before it wears out. To dump the star figure (and the text with it), the hammer had to hit the daisy wheel leaf with the period 1,423 times. My *Silver-Reed Owner's Manual* did not give any clues as to how long I can expect to use this same print wheel, but if the condition of the wheel (I have examined it under a microscope) is any indication, I can expect to go on using it for a long time. To date, I have done about 30 screen dumps with it, each averaging about 2,000 hits. How many screen dumps *your* printer can stand will depend upon how dense the dump is, how often you will be using it for graphics and how well the wheel is made. Just check it often for any signs of unusual wear.

You can create some unusual effects by using different HMI and VMI values along with the use of some character other than a period. For example, try using an asterisk (*) or a plus sign (+) instead of a period in Line 63100 of Listing 2. Play with the HMI and VMI values until you hit upon a combination you like.


Figure 1: Output of Listing 1

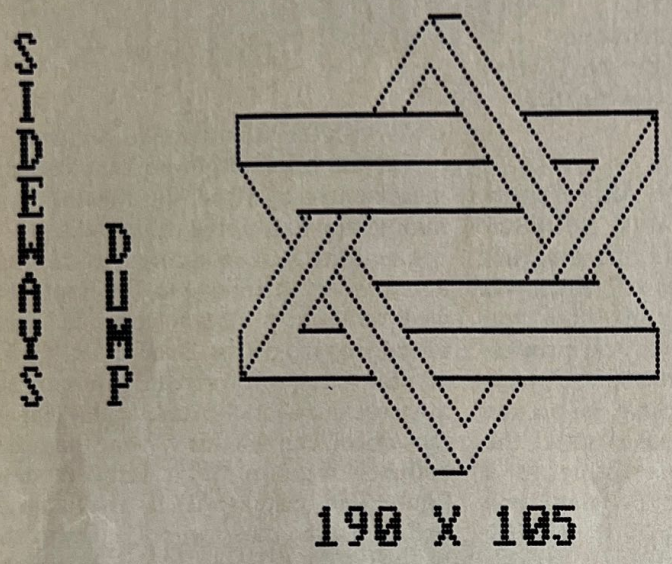



Figure 2: Sample Screen Dump

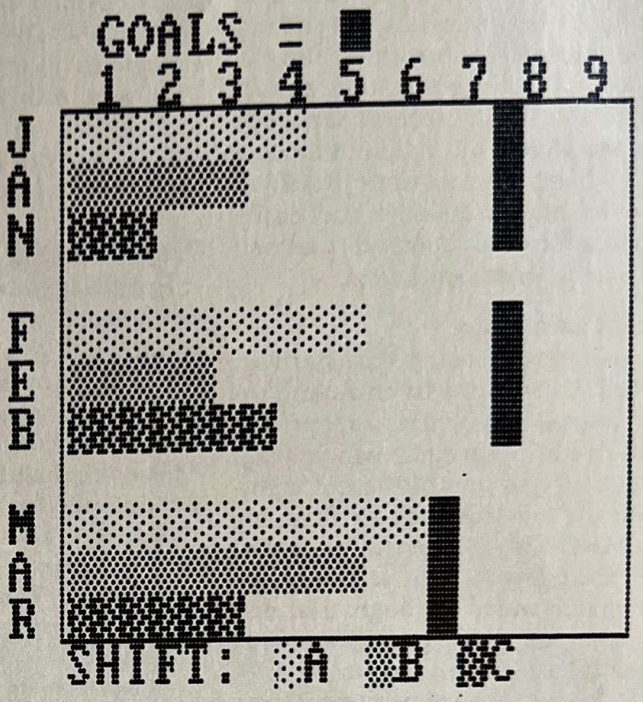



Figure 3: Example of Various Horizontal Motion Index Factors

An HMI of 5 did this.

This had an HMI of 4.

An HMI of 3 does this.

**Listing 1:** STARDUMP.BAS

```
5 TIME$="ØØ:ØØ:Ø1"
1Ø CLS:LOCATE 3,1,Ø
15 PRINT "S":PRINT "I":PRINT "D":PRINT "E":PRINT "W  D":PRINT "A  U":PRINT "Y  M
":PRINT "S  P"
16 PRINT :PRINT
2Ø FOR N=1 TO 36
3Ø READ X1,Y1,X2,Y2:Y1=Y1-1Ø:Y2=Y2-1Ø
4Ø LINE (X1,Y1)-(X2,Y2),3
5Ø NEXT N
6Ø DATA 112,24,12Ø,24,112,24,146,58,12Ø,24,15Ø,54,112,24,96,4Ø,1Ø4,4Ø,112,32,112
```

```
,32,142,62
7Ø DATA 136,4Ø,172,4Ø,172,4Ø,144,68,172,4Ø,172,48,172,48,152,68,144,48,156,48,15
6,48,136,68
8Ø DATA 158,62,172,76,172,76,172,84,172,76,1Ø4,76,172,84,112,84,154,66,156,68,15
6,68,96,68
9Ø DATA 128,84,12Ø,92,12Ø,92,9Ø,62,136,84,12Ø,1ØØ,12Ø,1ØØ,86,66,12Ø,1ØØ,112,1ØØ,
112,1ØØ,82,7Ø
1ØØ DATA 88,76,76,76,76,76,96,56,96,84,6Ø,84,6Ø,84,88,56,6Ø,84,6Ø,76,6Ø,76,8Ø,56
11Ø DATA 12Ø,4Ø,6Ø,4Ø,6Ø,4Ø,6Ø,48,6Ø,48,128,48,6Ø,48,74,62,78,58,76,56,76,56,136
,56
12Ø DATA 1,112,28,154,66,7Ø,7Ø,3,128.36,148,72,82,74,5,112,36,8Ø,68,128,8Ø
13Ø DATA 2,1ØØ,52,15Ø,5Ø,1ØØ,84,6,78,58,1Ø4,8Ø,16Ø,56,7,1ØØ,44,12Ø,88,16Ø,44
135 PRINT "          19Ø X 1Ø5"
136 PRINT :PRINT :PRINT
14Ø GOSUB 1ØØØØ
145 PRINT TIME$
15Ø STOP
1ØØØØ WIDTH 2ØØ
1ØØ1Ø LPRINT CHR$(27);CHR$(31);CHR$(5)
1ØØ2Ø LPRINT CHR$(27);CHR$(3Ø);CHR$(2)
1ØØ3Ø FOR HORIZ=Ø TO 19Ø
1ØØ4Ø FOR VERTI=Ø TO 1Ø5
1ØØ6Ø IF COLOR(HORIZ,VERTI)<>Ø THEN C$="." ELSE C$=" "
1ØØ7Ø A$=C$+A$
1ØØ8Ø NEXT VERTI
1ØØ9Ø LPRINT A$:A$=""
1Ø1ØØ NEXT HORIZ
1Ø11Ø RETURN
```

**Listing 2:** GRAPHICS.BAS — **Daisy wheel screen dump subroutine**

```
62995 REM Puts X aand Y cursor lines on screen, intersect at 3ØØ,1ØØ
63ØØØ GCURSOR(3ØØ,1ØØ),(BX,BY)
63ØØ5 REM Defines upper left corner of screen print box
63Ø1Ø FIRSTX=BX:FIRSTY=BY
63Ø15 REM Puts X and Y cursor back on screen
63Ø2Ø GCURSOR(BX,BY),(BX,BY)
63Ø25 REM Defines for subprogram lower right corner of box
63Ø3Ø SECONDX=BX:SECONDY=BY
63Ø45 REM Sets printer line width
63Ø5Ø WIDTH 2ØØ
63Ø55 REM Sets HMI (1/12Ø inch X (n-1)), (n is 5 here)
63Ø6Ø LPRINT CHR$(27);CHR$(31);CHR$(5)
63Ø65 REM Sets VMI (1/48 inch X (n-1), (2 is minimum)
63Ø7Ø LPRINT CHR$(27);CHR$(3Ø);CHR$(2)
63Ø75 REM Incremental movement on X axis
63Ø8Ø FOR HORIZ=FIRSTX TO SECONDX
63Ø85 REM Incremental movement on Y axis for building A$
63Ø9Ø FOR VERTI=FIRSTY TO SECONDY
63Ø95 REM Checks to see if color is black or not, assigns . or space to C$
631ØØ IF COLOR(HORIZ,VERTI)<>Ø THEN C$="." ELSE C$=" "
631Ø5 REM Builds A$ from C$ periods or spaces
6311Ø A$=C$+A$
6312Ø NEXT VERTI
63125 REM Prints A$
6313Ø LPRINT A$:A$=""
6314Ø NEXT HORIZ
6515Ø RETURN
```

# Two Great Holiday Gift Ideas


## A SOFT SECTOR GIFT CERTIFICATE

Give your Sanyo friends a gift you know they'll want under their tree — a year's subscription to SOFT SECTOR.
First, they'll receive a handsome card announcing your gift. Then your friends will be reminded of you each month for the next 12 months when they receive a four-color glossy issue of SOFT SECTOR. Each issue includes the information, programs, reviews and all important technical advice that you've come to value.

When you give SOFT SECTOR to your friends, you'll also be safeguarding your own collection. No more frantic searching for those back issues you've lent a friend.

Give a SOFT SECTOR gift certificate and let your friends in on something you already know — SOFT SECTOR is the perfect companion for the Sanyo MBC-550, 555 and 775 series computers!

*Get your order to us by Nov. 10 and we'll begin your friends' subscriptions with the January SOFT SECTOR.*

**Please begin a one-year (12 issues) gift subscription to SOFT SECTOR for:**

Name _____
Address _____
City _____ State _____ ZIP _____

**From:**
Name _____
Address _____
City _____ State _____ ZIP _____

☐ My payment is enclosed
☐ Please bill to my ☐ VISA ☐ MasterCard ☐ American Express
My account # _____ Exp. date _____
Signature _____

Subscriptions to SOFT SECTOR are $28 in the United States, U.S. $35 in Canada. The surface rate to other countries is U.S. $64; the air rate, U.S. $85. Kentucky residents add 5% sales tax. **U.S. currency *only*, please. All subscriptions begin with the current issue.** Please allow 5 to 6 weeks for delivery. In order to hold down costs, we do not bill.

Mail to:
**SOFT SECTOR Gift Certificate, The Falsoft Building, Prospect, KY 40059.**
To order by phone, call (502) 228-4492

## DISTINCTIVE AND DURABLE SOFT SECTOR BINDERS

Do yourself and your SOFT SECTOR friends a favor! Protect and showcase a valuable resource — each and every issue of SOFT SECTOR — with high-quality, blue vinyl binders with the magazine's name embossed in gold.

Spend more time with your Sanyo and less on frustrating searches for misplaced issues. Organize your SOFT SECTOR library!


These handsome royal blue binders cost just $7.50 (please add $2.50 for shipping and handling per binder).

**SAVE!** Buy four or more back issues of SOFT SECTOR with this order and save $1 per issue. Use the Back Issue Order Form on Page 51.

Mail to:
**Soft Sector Binders**
**The Falsoft Building**
**Prospect, KY 40059**

☐ **YES. Please send me**_____ **SOFT SECTOR binder(s)** at $7.50 each (plus $2.50 for shipping and handling per binder). If your order is to be sent via U.S. Mail to a post office box or to another country, please add $2. Kentucky residents add 5% sales tax.

Name _____
Address _____
City _____ State _____ ZIP _____
☐ My check in the amount of _____ is enclosed. (In order to hold down costs, we do not bill.)
Charge to: ☐ VISA ☐ MasterCard ☐ American Express
Account Number _____ Exp. Date _____
Signature _____

# SOFT TALK

**PRODESIGN II** is a computer-aided design system which allows you to make drawings of all sorts from engineering to artwork. Its features and capabilities compare to other CAD systems and it can be used by both the novice and professional.

*ProDesign II* requires an MBC-775 or a Video RAM Board and 512K. It is currently available for $299, $3 S/H from American Small Business Computers, 118 So. Mill Street, Pryor, OK 74361; (918) 825-4844.

* * *

**E-Z-DOS-IT** is a program which allows DOS to do more than one task at a time. You can run up to eight tasks at the same time depending on the size of your computer's memory and the memory requirements of the applications programs you select.

*E-Z-DOS-IT* requires a Video RAM Board or an MBC-775 and 256K. It is currently available for $199.95 from Hammer Computer Systems, Inc., 700 Larkspur Landing Circle, Suite 285, Larkspur, CA 94939; (415) 461-7633.

* * *

**REFER** is a program which helps save programmers time by processing any microcomputer language. As it reads your program, it prints a source code listing. This program also concurrently analyzes your program and prints a cross-reference of variables, values, keywords, line numbers and/or labels.

*Refer* can help you when you write, test, document and change programs. It comes with compiled code, source code and examples.

*Refer* is currently available for $39 from James Halstead & Associates, 1551 Plainfield Road, Joliet, IL 60435; (815) 725-0346.

* * *

**OPEN ACCESS** is a software package that integrates a relational database, a spreadsheet, 3-D business graphics, word processing, time management and address file, telecommunications and conversion utilities for SIF, ASCII text and DBASE files.

*Open Access* is entirely menu driven. From the main menu you can enter any of the modules, access the program's utilities section or exit to the operating system. This program is also available in German, French, Spanish, Dutch, Italian, Swedish, Norwegian, Finnish and Portuguese.

*Open Access* requires a Video RAM Board, an MBC-775 and 256K. It is currently available for $695, $3 S/H from Software Products International, 10240 Sorrento Valley Road, San Diego, CA 92121; (619) 450-1526.

* * *

**BOOK SENSE.** The *Sanyo BASIC User's Handbook* is a guide to Sanyo BASIC for the MBC-550/555 series computers. For beginners, this book offers step-by-step instructions in the use of Sanyo BASIC. For experienced programmers, there are discussions of graphics, file handling, machine language calls and I/O.

The *Sanyo BASIC User's Handbook* is currently available for $17.95 from Weber Systems, Inc., 8437 Mayfield Road, Chesterland, OH 44026; (216) 729-2858.

* * *

**SANYGRAF** is a business and scientific graphics package. It will transform numerical reports from your spreadsheet, database or application programs to understandable line graphs, and bar or pie charts of your data in full color.

*SanyGraf* requires MS-DOS 2.11 and at least 256K. It is currently available for $99 from A-OK Computers, 816 Easley St., Suite 615, Silver Spring, MD 20910; (301) 585-5105.

* * *

*The products and services listed above are currently available for Sanyo personal computers and many will be reviewed in future issues of SOFT SECTOR.*

# ONE-LINERS

*Here we are again with more winners of The First Great SOFT SECTOR One-Liner Contest. The authors of these winning entries will receive a copy of the October edition of SOFT SECTOR ON DISK.*

*To briefly restate the rules of the contest, begin the program with the number one and end it in a single line. Anything else goes. Entries will be accepted in either Sanyo BASIC or GW-BASIC. Include a printed listing, a title for the program and a short explanation of what it does. Send it to The First Great SOFT SECTOR One-Liner Contest, P.O. Box 385, Prospect, KY 40059.*

★★★★★

## BLUE STAR

This program will begin by clearing the screen and then draw a star using a line design technique. When you type in this program you should type it in exactly as it is written so you will be sure to get it all in on one line. I am 13 years old and I have been enjoying my Sanyo since Christmas.

```
1 CLS:FOR X=3ØØ TO 48Ø STEP 1Ø:LINE(3ØØ,Y)-(X,9Ø):Y=
Y+5:NEXT:FOR X=47Ø TO 3ØØ STEP-1Ø:LINE(3ØØ,Y)-(X,9Ø)
:Y=Y+5:NEXT:FOR X=29Ø TO 13Ø STEP-1Ø:LINE(3ØØ,Y-1Ø)-
(X,9Ø):Y=Y-5:NEXT:FOR X=12Ø TO 29Ø STEP 1Ø:LINE(3ØØ,
Y-1Ø)-(X,9Ø):Y=Y-5:NEXT:FOR T=1TO X*5Ø:NEXT
```

*Kyle C. Quinnell*
*Las Cruces, NM*

★★★★★

## SLOT MACHINE

This program asks you if you want to play. It then clears the screen, draws the slot machine and gives your first selection. It even congratulates you when you win!

```
1 INPUT "PLAY";A$:IF A$="N" THEN END ELSE CLS:FOR P=
1 TO 3:LINE(99-P,4Ø-P)-(465+P,135+P),P,B:D(P)=INT(RN
D*3+1):SYMBOL(P*99+P*25,5Ø),CHR$(D(P)),9,9:NEXT:IF D
(1)<>D(2) OR D(1)<>D(3) THEN 1 ELSE FOR S=1 TO 7:CLS
:BEEP:SYMBOL(7Ø,5Ø),"BRAVO !",S,8:BEEP:NEXT
```

*Yvan Fortin*
*Montreal PQ Canada*

## SUBSCRIPTION INFORMATION

Your copy of SOFT SECTOR is sent second class mail. If you do not receive your copy by the 15th* of the month of the cover date, send us a card and we will mail another. *Canadian subscribers and foreign air mail allow two additional weeks.

You must notify us of a new address when you move. Notification should reach us no later than the 10th of the month prior to the month in which you change your address. Sorry, we cannot be responsible for sending another copy when you fail to notify us.

Your mailing label also shows an "account number" and the subscription expiration date. Please indicate this account number when renewing or corresponding with us. It will help us help you better and faster.

For Canadian and other non-U.S. subscribers, there may be a mailing address shown that is different from our editorial office address. Do not send any correspondence to that mailing address. Send it to our editorial offices at P.O. Box 385, Prospect, KY 40059.




# BUSINESS SECTOR


CHARLOTTE & BRIAN STONE
Soft Sector Contributing Editors


**Q.** *I have a Tandy DMP-105 printer that I would like to use with my Sanyo MBC-555. Can you furnish any information on how to install the printer with the software that comes with the computer? Also, I get a second linefeed every time. Can this printer do graphics?*

*William F. Steagall, Sr.*
*Santa Monica, CA*

**A.** After studying the manual pages sent with your second letter, I think the following will help with using your DMP-105 with *WordStar*.

Reinstall a copy of *WordStar* for a "Standard Printer," from the printer menu. After you have done the installation, change the printer initialization to the following Hex codes

# SUBMITTING MATERIAL

Contributions to SOFT SECTOR are welcome from everyone. We like to run a variety of programs which will be useful/helpful/fun for other Sanyo owners.

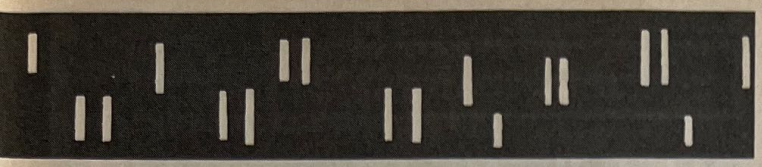
• **FORMAT:** Unless the program accompanying your submission is less than 10 lines, we must have the program itself on disk. We will print out the listing to our specifications. We simply cannot take the time to key in (and debug our typing errors) material which is longer than that. Editorial copy can also be included on disk, using any of the word processors currently available for the Sanyo 550, 555 or 775. However, please also include a double-spaced hard copy of your editorial material and hard copy of your program listing. Please do not send text in all capitals. Use upper- and lowercase. While it is a big help to us in typesetting for you to send your article saved on disk using the ASCII option, it is not mandatory. But we must have, at the very least, a double-spaced hard copy of the article.

• **WHAT TO WRITE:** Anything with a practical application. If it interests you, it will probably interest a lot of others. However, we vastly prefer articles with accompanying programs which can be entered and run. The more unique the idea, the more appeal. We can prepare finished tables, diagrams and schematics from your rough draft *if* you provide legible copy and full directions. We have a continuing need for short articles with short listings.

We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those who wish more detailed information on making submissions, please send an SASE to: Submissions Editor, SOFT SECTOR, P.O. Box 385, Prospect, KY 40059. We will send you some more comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.



*(Charlotte Stone, office manager for the Detroit office of the Shaw/Walker Co., has been using a Sanyo computer in her daily work routine since October of 1983. Brian Stone has been using a variety of Sanyo computers since May of 1983. Charlotte and her husband, Brian, have been involved with computers since July of 1978 and presently own and use five computer systems on a daily basis, three of which are Sanyos.)*

— 1Bh and 15h. This will hopefully tell the printer *not* to send a linefeed with a carrige return. This is what *WordStar* expects the printer to do and this is why you are getting the double-spaced text.

I do not know how to help you generate graphics with the printer because it is designed to generate TRS-80 graphics rather than Epson-style graphics, which most software will try to do. I hope this is of some help for your *WordStar* use.

---

**Q.** *Can you tell me if a Sanyo MBC-555 with the Video RAM Board will run the screen editor of* dBase II*?*

*George F. Jenkins*
*Bloomington, MN*

**A.** I have checked with one of my sources and he thinks *dBase II* should run without modification on a machine with the Video RAM Board.

---

**Q.** *I would like to insert accent marks when entering data with* InfoStar *and would like to print more than one copy of a file with* WordStar. *I don't want to merge anything. How do I do these things?*

*Mickey Raymer*
*Ava, MO*

**A.** In regard to using *MailMerge*, just answer the questions like you would if you were just printing a letter with the 'P' option. The prompt "Name of file to merge" is just asking for the filename of what you want to print. You will finally get to the question "How many copies to print?"

From what we can find, you will not be able to insert accent marks in the variables when using *InfoStar*. Most of the *WordStar* controls can be used to emphasize report headings but not data which you will input.

---

**Q.** *I have a TI-855 dot matrix printer and would like to use it with* WordStar *but do not know how to install it because it is not on the menu.*

*Pat Callahan*
*La Crosse, WI*

**A.** I am sorry to have taken so long to respond to your letter. One reason for the delay is because I had trouble finding out anything about your printer.

I have found that your printer is supposed to emulate an Epson in draft mode and a Diablo in letter-quality mode. The printer probably defaults to TI codes and must be toggled to get the other two modes. Have you tried this?

Try installing *WordStar* as the Epson choice from the printer install menu and see what it will do, then install another copy as the Diablo 630, then the 1610 and see which of these works best with your printer.

---

**Q.** *I would like to find a patch to* CalcStar *to allow the use of* all *128K of memory for the spreadsheet without upgrading my RAM. I have been told by Sanyo that without the memory upgrade I can expect only about 7K of memory.*

*David E. Ritter*
*Pottstown, PA*

**A.** There is no patch to allow *CalcStar* to use all of the 128K of memory in a 128K computer. The operating system uses about 30K of the RAM and the program itself is a little smaller than 54K. If we take a look at some of the figures involved, you will see why your spreadsheets are so small.

| | | |
|---|---|---|
| 128K machine = | 131,072 bytes of RAM without DOS loaded. | |
| | 128K machine = | 77,888 bytes of RAM with DOS 2.11 and system uses. |
| Subtract *CalcStar* | 54,000 | |
| RAM available for spreadsheet | 23,888 | You must also subtract room for the overlays. |

---

**Q.** *I would like to run* Lotus 1-2-3, MultiMate *and* Personal Editor *on my MBC-550. What hardware or software considerations should be observed?*

*George E. Orient*
*Los Angeles, CA*

**A.** The only way to get the Sanyo MBC-550 series to closely emulate the IBM PC is to have the Video RAM Board installed. This will guarantee that *Lotus 1-2-3* will run, but I do not know for sure if *MultiMate* or *Personal Editor* will also run.

Your dealer will be able to get the Video RAM Board. It retails for $150 and can be installed by the user. A new version of MS-DOS and a version of GW-BASIC is included.

---

**Q.** *I would like to find an idea processor or an outliner for the Sanyo MBC-550. Are there any on the the market*

*and are they compatible? Also, how do I determine compatibility of IBM programs for use with MS-DOS 2.11?*

*Rev. William B. Curtin, OP*
*San Francisco, CA*

**A.** The only idea processor that I am aware of and which *may* work, is a program called *Think Tank* for the IBM PC. The reason I say may work is because I think that it is copy protected and thus may not be able to be copied to the Sanyo MS-DOS. Some copy protected software must boot and an IBM version will not do so in the MBC-550 series.

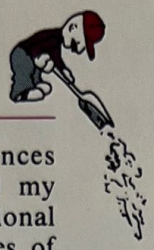
Regarding the compatibility of programs for the Sanyo, there is never a guarantee that IBM software will run. It is not the DOS which is not compatible; it is the hardware (the computer itself). The only way to get greater odds on your side is to have the Video RAM Board installed in the machine. This *does not* guarantee total compatibility because you still will not have an IBM, but it does allow more programs to function.

---

**Q.** *How can I get my Telex Daisy Printer to do super- and subscript? It seems that I would have to install* WordStar *for a different printer but I do not know what choice to make from the menu.*

*Michael Doeff*
*El Cerrito, CA*

**A.** I am not sure which daisy wheel printer your Telex Daisy Printer emulates, but you could install a copy of *WordStar* for the Diablo 1610 and 630 printers to see if one of those drivers will work correctly. According to the copy of the manual that you sent, your printer will super- and subscript. If one of the Diablo selections does not work, your only choice is to try each of the daisy wheel choices on the menu until you find one that works. My best guess is the Diablo because most manufacturers use those codes.

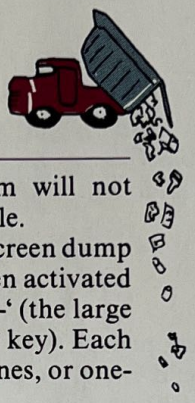


# Down In T

## ASCII Sc

## f

## Sanyo ME



*By Rev. M*

There are enough differences between an IBM PC and my Sanyo to give me an occasional dose of envy. One of the features of "Big Blue" that I often wished I had was a character screen dump.

There are a couple of ways to get around this. The Sanyo allows you to press CONTROL-P and echo the screen output to the printer. There are times, however, when I don't know that I want something printed until I see it on the screen.

Sanyo does provide a graphics screen dump that I could use, but that solution is slow, incomplete and not really what I want. The unaltered Epson graphics screen dump prints only 480 of the 640 horizontal pixels on the eight-inch carriage of my Panasonic 1091. I could patch the program's assembly language source code so that it prints 640 columns on my printer (in fact, I have done so), but this process is still slower, harder on the print head and ribbon, and less "businesslike" than an ordinary character screen dump program would be.

So, I decided to write my own screen dump program. There are commercial screen dump programs available, but one of the reasons I bought a Sanyo is that I hate to spend money. Plus, I like the adventure of exploring the inner workings of my computer. I have never before programmed in assembly language, so this program is testimony to what a determined user can do with a little help.

**Operating Instructions**

For those who don't have access to an assembler, or who don't care about learning assembly language, the BASIC program listed below will produce the ASCII-SD.COM file. Those who assemble and link the program should also run EXE2BIN. The program will not work properly as an .EXE file.

The .COM file installs the screen dump program in memory. It is then activated by pressing CONTROL-SHIFT-' (the large key to the left of the RETURN key). Each screen print will require 33 lines, or one-half of an ordinary page.

It should be noted that the .COM program must be run each time the computer is turned on or reset, in order for the screen dump to be present. The converse is also true. The program need only be run once each session. Once the code is installed in memory, it stays there until you turn the power off or press Reset. If you always want the screen dump installed, this can be accomplished automatically with an AUTOEXEC.BAT file.

No harm is done if you try to install the program more than once. The screen dump routine can also be called from a user program by calling INT 05. Those interested in how it works should read on.

It was SOFT SECTOR that gave the clues I needed to write the program. In the February 1985 issue, Page 86, Tim Purves mentioned that Function 71h of Interrupt 10h returned a pointer to the character RAM. That was an essential clue. Richard Kruse's article (SOFT SECTOR, June 1985, Page 10) on the organization of the Sanyo's video RAM also provided essential information. The Sanyo video system keeps in memory, a read-only copy of the character screen. This portion of RAM is 4,000 bytes long. Within these 4,000 bytes, the characters are arranged sequentially from first column to last column, and from first line to last line. Each character is represented by two bytes. The first byte represents the ASCII code of the character, and the second byte represents its video attribute (color or intensity). Finally, Larry Forman's letter in SOFT SECTOR (June 1985, Page 70) gave me some hints about how the print screen routine works in the IBM PC and the Sanyo Video RAM Board versions of DOS.

I studied this information, along with the technical reference section of the *Sanyo Operator's Guide* and a couple of books on IBM PC assembly language programming, and here's what I came up with. The program works much like the Epson graphics dump provided by Sanyo.

*(Mitchell Lewis is an ordained elder in the United Methodist Church. He purchased his Sanyo MBC-555 to use in his ministry and he assists congregations and other units of the church with their computer setups and operations. Mitchell can be contacted c/o the New Pentecost United Methodist Church, Route 4, Box 122, Winder, GA 30680.)*

# ...e Dumps

## ...een Dump

## ...r

## ...C 550/555

...chell Lewis

When any key is pressed, the 8259 Interrupt Controller alerts the 8088 microprocessor of that fact. The 8088 then temporarily stops what it is doing to process the data from the keyboard. It does this by calling Interrupt 251 (FB Hex).

The processor looks for the address of all interrupt service routines in Segment 0000 at an offset of four times the number of the interrupt. In our case, the segment and offset of the routine which process the keyboard input are located in segment 0000 at an offset of 4 * 251 (or 1004). These addresses of the interrupt service routines are known as "vectors."

The program begins by patching the interrupt vectors. It patches the vector of the the original keyboard service routine into an ordinarily unused interrupt — number 247 (F7 Hex). It then causes the vector of Interrupt 251 to point to the program's own keyboard service routine.

Finally, the program patches the vector of the actual screen dump routine into the address for Interrupt 5. Interrupt 5 is unused in the bundled version of MS-DOS, but points to a screen dump routine in PC-DOS and Sanyo's MS-DOS 2.11 VB. Since the vector at INT 5 now points to our screen dump routine, we can call it from user programs without requiring keyboard input. Programs that utilize this feature should no longer result in an "Interrupt Trap Halt" with the bundled version of the Sanyo DOS.

After the patches are complete, the program returns control to DOS. The working portions of the program (the screen dump routine and the keyboard service routine) remain in memory, while the memory used for the installation code is released for other programs.

Now, when a key is pressed, the computer jumps to the portion of code labeled INT_FB. The code first does the normal keyboard service routine (now INT F7 Hex) and then checks to see if the key pressed was CONTROL-SHIFT-'. If so, the program calls Interrupt 5 (the screen dump) and clears the keyboard buffer when finished.

The screen dump itself works in this order:

1) Save the contents of the 8088's registers.
2) Disable the keyboard from causing another interrupt while the screen dump is in progress. This is done by masking out the IR3 pin of the 8259 Interrupt Controller. Page 5-18 of the *Sanyo Operator's Guide* tells what each of the eight pins of 8259 is used for. The byte at I/O Port 02 indicates whether the 8259 is allowed to forward the interrupt request or not. Each bit represents a pin on the 8259. The program fetches the current mask from Port 02, forces the 8's bit to one (xxxx1xxx), and sends that byte back out to Port 02.
3) Find the address of the character screen memory. Tim Purves showed us how to use BIOS call 10 Hex to fetch the address of the character screen memory. We use the ES segment to refer to the character screen segment, and the SI register to point to the offset of the first byte of the character screen memory.
4) Print the 25 lines of the screen, followed by eight blank lines. This makes a total of 33 lines, or one-half of a normal page. Notice that these routines use loops. The CX register is the loop counter. Before executing the loop, CX is set to the desired number. When the LOOP command is encountered, CX is decremented by one and control passes back to the address indicated by the following label. This portion of the program makes use of subroutines labeled PRT1LN, DOSPRN and LNFEED. PRT1LN saves the line count, prints 80 characters and restores the line count. The character to be printed is located at the address ES:SI. These are the registers we first set when we fetched the address of the character screen. Each time a character is printed, SI is incremented by two. This skips over the attribute byte associated with the character just printed and makes SI point to the next character. DOSPRN uses the BIOS routine 17 Hex, described on Page 5-21 of the *Sanyo Operator's Guide*.
5) Clear the keyboard buffer of all keys pressed during the screen dump. The program does this by reading the keyboard input port on the 8088. Page 5-27 of the *Sanyo Operator's*

*Guide* gives the address of these ports; 38 Hex for data and 3A Hex for Command/Status. The keyboard I/O is done with an 8251 USART, just like the one that controls serial I/O through the RS-232C. Page 5-30 of the *Sanyo Operator's Guide* that shows the second bit from the right (the 2's bit) indicates the presence of data to be processed. The CLRBUF routine gets the contents of the Command/Status port (3A Hex) and checks to see if the 2's bit is high. If it is, the

routine reads in a character from port 38 Hex and sends an error reset to the USART in case the CONTROL key is pressed. The Sanyo CONTROL key causes the keyboard data to be sent with a parity error (*Sanyo Operator's Guide* Page 5-32). I don't really know the size of the keyboard buffer so I just loop until it is empty.

6) Allow the 8259 to pass along keyboard interrupts once again by forcing the 8's bit of the interrupt mask to zero.
7) Restore the registers to their original contents.
8) Return to the calling program.

I've learned a lot from SOFT SECTOR. I hope this program is both educational and useful for you.

**Listing 1:** ASCII-SD.BAS

```
1ØØ DATA E9,87,ØØ,FA,5Ø,53,51,52,56,Ø6,1E,ØE,1F,E4,Ø2,ØC,Ø8,E6,Ø2,FB
11Ø DATA B4,71,CD,1Ø,1E,8E,D8,8E,Ø7,8B,D9,8B,37,1F,B9,19,ØØ,E8,27,ØØ
12Ø DATA E2,FB,B9,Ø8,ØØ,E8,3B,ØØ,E2,FB,FA,E4,3A,A8,Ø2,74,Ø8,E4,38,BØ
13Ø DATA 35,E6,3A,EB,F2,E4,Ø2,24,F7,E6,Ø2,1F,Ø7,5E,5A,59,5B,58,CF,51
14Ø DATA B9,5Ø,ØØ,26,8A,Ø4,E8,ØA,ØØ,83,C6,Ø2,E2,F5,E8,ØA,ØØ,59,C3,B4
15Ø DATA ØØ,BA,ØØ,ØØ,CD,17,C3,BØ,ØD,E8,F3,FF,BØ,ØA,E8,EE,FF,C3,CD,F7
16Ø DATA 5Ø,B4,Ø1,CD,16,3D,1E,Ø7,75,Ø6,CD,Ø5,B4,ØØ,CD,16,58,CF,FA,33
17Ø DATA CØ,8E,CØ,26,A1,EC,Ø3,3D,76,Ø1,74,2A,26,A3,DC,Ø3,26,A1,EE,Ø3
18Ø DATA 26,A3,DE,Ø2,B8,76,Ø1,26,A3,EC,Ø3,26,8C,ØE,EE,Ø3,B8,Ø3,Ø1,26
19Ø DATA A3,14,ØØ,26,8C,ØE,16,ØØ,BA,8B,Ø1,FB,CD,27,B4,ØØ,CD,21
2ØØ CLS
21Ø PRINT "Creating ASCII Screen Dump 'ASCII-SD.COM'.  Please Stand by"
22Ø OPEN "O",#1,"ASCII-SD.COM"
23Ø FOR I%=1 TO 198
24Ø READ A$
25Ø PRINT #1;CHR$(VAL("&H"+A$));
26Ø PRINT ".";
27Ø NEXT I%
28Ø CLOSE #1
29Ø PRINT:PRINT "ASCII Screen Dump 'ASCII-SD.COM' has been created"
```

**Listing 2: The assembly language source code**

```
TITLE   ASCII SCREEN DUMP FOR SANYO MBC-55Ø/555
CSEG    SEGMENT
        ORG     1ØØH
        ASSUME  CS:CSEG, DS:CSEG
START:  JMP     INIT

INT_Ø5: CLI
        PUSH    AX              ;SAVE REGISTERS
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    SI
        PUSH    ES
        PUSH    DS
        PUSH    CS              ;SET DS := CS
        POP     DS

        IN      AL,2            ;GET 8259 INTERRUPT MASK
        OR      AL,8            ;MASK OUT KEYBOARD (IR3)
```

```
        OUT     2,AL
        STI

        MOV     AH,71H          ;GET POINTER TO CHARACTER SCREEN
        INT     1ØH
        PUSH    DS
        MOV     DS,AX
        MOV     ES,[BX]
        MOV     BX,CX
        MOV     SI,[BX]
        POP     DS

        MOV     CX,25           ;SET COUNTER TO 25 LINES
NEXTLN: CALL    PRT1LN          ;PRINT ONE LINE
        LOOP    NEXTLN
        MOV     CX,8            ;SET COUNTER TO 8 LINES
NEXTLF: CALL    LNFEED          ;ISSUE LINE FEED
        LOOP    NEXTLF          ;(MAKES EVEN 1/2 PAGE)
        CLI

CLRBUF: IN      AL,3AH          ;CLEAR KEYBOARD BUFFER
        TEST    AL,2            ;OF ALL INPUT
        JZ      KEYSON
        IN      AL,38H
        MOV     AL,35H
        OUT     3AH,AL
        JMP     CLRBUF

KEYSON: IN      AL,2            ;RE-ENABLE KEYBOARD
        AND     AL,ØF7H         ;UN-MASK INTERRUPT IR3
        OUT     2,AL

        POP     DS              ;RESTORE REGISTERS
        POP     ES
        POP     SI
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        IRET                    ;RETURN FROM INT Ø5 CALL

PRT1LN: PUSH    CX              ;SAVE COUNT OF LINE NUMBER
        MOV     CX,8Ø           ;SET COUNTER TO 8Ø COLUMNS
NEXTCH: MOV     AL,ES:[SI]
        CALL    DOSPRN          ;PRINT ONE CHARACTER
        ADD     SI,2            ;INCREMENT INDEX
        LOOP    NEXTCH
        CALL    LNFEED          ;ISSUE LINE-FEED WHEN DONE
        POP     CX              ;RESTORE LINE NUMBER TO CX
        RET

DOSPRN: MOV     AH,Ø            ;BIOS PRINTER ROUTINE
        MOV     DX,Ø            ;SEE SANYO USERS MANUAL
        INT     17H
        RET

LNFEED: MOV     AL,ØDH          ;CARRIAGE RETURN
        CALL    DOSPRN
        MOV     AL,ØAH          ;LINE-FEED
```

```
        CALL    DOSPRN
        RET

INT_FB: INT     247             ;FIRST DO NORMAL KEYBOARD INT
        PUSH    AX
        MOV     AH,1            ;LOOK AT CHARACTER IN BUFFER
        INT     16H
        CMP     AX,Ø71EH        ;THE CTRL-SHIFT-` CHARACTER?
        JNE     GO_ON
        INT     5H              ;DO SCREEN DUMP
        MOV     AH,Ø
        INT     16H             ;CLEAR KEYBOARD BUFFER
GO_ON:  POP     AX
        IRET                    ;RETURN FROM INT FB CALL

INIT:   CLI
        XOR     AX,AX                   ;PATCH OLD INT FB VECTOR
        MOV     ES,AX                   ;INTO INTO UNUSED VECTOR
        MOV     AX,ES:[ØFBH*4]          ;OF INT F7
        CMP     AX,OFFSET INT_FB
        JE      ABORT                   ;ABORT IF ALREADY INSTALLED
        MOV     ES:[ØF7H*4],AX
        MOV     AX,ES:[ØFBH*4+2]
        MOV     ES:[ØF7H*4+2],AX

        MOV     AX,OFFSET INT_FB        ;PATCH THE ADDRESS OF THIS
        MOV     ES:[ØFBH*4],AX          ;PROGRAM'S KEYBOARD ROUTINE
        MOV     ES:[ØFBH*4+2],CS        ;INTO THE VECTOR AT INT FB

        MOV     AX,OFFSET INT_Ø5        ;PATCH ADDRESS OF SCREENDUMP
        MOV     ES:[Ø5H*4],AX           ;ROUTINE INTO UNUSED VECTOR
        MOV     ES:[Ø5H*4+2],CS         ;OF INT Ø5

        MOV     DX,OFFSET INIT+1        ;EXIT AND STAY RESIDENT
        STI                             ;RELEASE UNNEEDED MEMORY
        INT     27H                     ;FROM THIS PROGRAM

ABORT:  MOV     AH,Ø                    ;JUMP HERE IF ALREADY INSTALLED
        INT     21H

CSEG    ENDS
        END     START
```

# RACKSELLERS

The stores listed below carry **SOFT SECTOR** on a regular basis and may have other products of interest to Sanyo 550 series users. We hope you will patronize those in your area.

**ALABAMA**
- Birmingham — The Computer Store
- Madison — Madison Books
- Mobile — Mall Tel
- Montgomery — Trade-N-Books

**ARIZONA**
- Flagstaff — Datasystems
- Phoenix — Gemini Computers
- Phoenix — Tri-Tek, Inc.
- Tempe — Books, Etc.
- Tempe — Computer Library

**CALIFORNIA**
- Folsom — Computers, Etc.
- Hayward — 3E Software & Systems
- Huntington Beach — Hammer Industries
- La Jolla — Affordable Computers
- Lancaster — Desert Computing
- Los Angeles — OPAMP
- N. Hollywood — Levity Distributors
- Norco — Wiest Custom Studios
- Riverside — Famis Computers
- Sacramento — Software Centre International
- San Bruno — Home Computing Centers, Inc.
- San Diego — Byte & Floppy Computers
- San Diego — Programs Unlimited
- San Luis Obispo — El Corral Bookstore
- San Luis Obispo — Paradise Computer Systems
- Santa Maria — Information Systems Unlimited
- Santa Rosa — Sawyer's News, Inc.
- Sherman Oaks — Data Systems West
- Stockton — Hardings Way News
- Upland — Computer Haven
- Upland — Friendly Computers
- Walnut Creek — ECX Computer Co.

**COLORADO**
- Aurora — Devi Computers
- Colorado Springs — The Computer Farm
- Englewood — Whole Life Distributors
- Security — Video Communications & Computers, Inc.

**CONNECTICUT**
- Manchester — Computer Training Institute
- Meriden — Total Computers

**DELAWARE**
- Wilmington — Nor-Mar — The Smoke Shop

**FLORIDA**
- Altamonte Springs — Micro's Etc.
- Boca Raton — Software, Software, Inc.
- Brooksville — All American Electronics
- Clearwater — Southern Micro Computer Co.
- Ft. Lauderdale — Computer Ease
- Gainesville — Computerized Applications
- Jacksonville — AA Computer Exchange
- Jacksonville — Florida Computer Resources
- Longwood — Electronic Specialty Products
- Melbourne — City News Stand
- Melbourne — B & B Office Equipment
- Merritt Island — Micro's, Etc.
- Miami — Computers, Computers, Computers
- Miami — Computer Image
- Orlando — Advanced Business Systems, Inc.
- Orlando — Rainbow Computer Center
- Ormond Beach — Micro's Etc.
- Point Ornge — Micro's Etc.
- Port Rickey — Computer Trends
- Sanibel Island — Sanibel Audio
- Sarasota — Family Computers
- Seminole — Home Computer Discount, Inc.
- Shalimer — Discount Discs
- St. Petersburg — Software Connection
- Tampa — Computer Dimensions
- Tampa — Personal Computer Center
- Tampa — Southern Micro Computer Co.

**GEORGIA**
- Augusta — Ram, Inc.
- Atlanta — The 64 Store
- Atlanta — Guild News Agency
- Atlanta — Southern Computer
- Columbus — Alboes Computers
- Lithonia — Certified, Inc.

**IDAHO**
- Boise — Western Computer & Office Supplies
- Moscow — H & S Computer Supplies
- Nampa — Canyon Computers & Communications

**ILLINOIS**
- Belleville — Software Or Systems
- Belleville — TPS Computers
- Chicago — Hyde Park Computers
- Salem — Salem Computer Systems

**INDIANA**
- Ft. Wayne — Bytrex
- Ft. Wayne — Digital Perspectives
- Indianapolis — Accolade Computer Design
- Kokomo — A Computer Store
- Kokomo — Toso Computer Shop

**KANSAS**
- Lawrence — Computer Outlet
- Wichita — Wichita Computer & Supply

**KENTUCKY**
- Cynthiana — Accounting Data Corp
- Louisville — Software Source
- Louisville — TCBC
- Owensboro — Computer Stall
- Prospect — Falsoft, Inc.

**LOUISIANA**
- Gretna — The Computer Supply Store
- Sulphur — Home Computer & Software
- Thibodaux — Great Southern Computer Systems

**MAINE**
- Bangor — Downeast Specialty Co.

**MARYLAND**
- Baltimore — Computers Edge
- Camp Springs — Compudeo
- Millersville — Comp-U-Type
- Pasadena — RW Electronics Computer Center
- Silver Springs — A-OK Computers
- Towsend — Greetings & Readings
- Waldorf — Waldorf Computer
- Wheaton — Micro Computer Co.

**MASSACHUSETTS**
- Danvers — Microcon Computers & Softwarecenter

**MICHIGAN**
- Allen Park — Galaxy Computers
- Ann Arbor — University Cellar
- Berkley — Family Computers
- Charlotte — Computer Options
- Dearborn — Alpha-K Computer Co.
- Dearborn — DSL
- Farmington — ComputerTowne
- Fenton — Computer Control Systems
- Fenton — Petry Products
- Grand Rapids — Bursma Electronic Dist. Co.
- Grand Rapids — Programs Unlimited
- Mt. Clemens — Computer Plus, Inc.
- Muskegon — 8-Bit Corner
- Owosso — C/C Computer Systems
- Pontiac — MichTron
- Rochester — Rochester Book Center
- Royal Oak — New Logic Computers
- Southgate — Popular Computer Centers, Inc.
- Sterling Heights — The Family Room
- Troy — Creative Center
- Wyandotte — ComputerTech
- Wyoming — Gerry's Book Co.
- Ypsilanti — Ye Olde Computer Shoppe

**MINNESOTA**
- Minneapolis — Computer Exercise World
- Rochester — Datalink

**MISSOURI**
- Aurora — Southland Electronics
- Columbia — Computer Concepts
- Creve Coeur — Alcor Associates
- Independence — Digital Enterprises, Inc.
- Republic — Franklin Computers
- St. Louis — Softwaire Centre
- St. Louis — Software Center

**MONTANA**
- Whitehall — Rambyte Computer Institute

**NEBRASKA**
- Omaha — Computers & Components

**NEVADA**
- Las Vegas — Computer Magic
- Reno — High-Tek Computer

**NEW JERSEY**
- Lake Hopatcong — Jefferson Computer Center
- Lawrenceville — Microcon Computer & Software
- Ocean City — The Computer & Software Store
- Westmont — American Computers

**NEW MEXICO**
- Alamogordo — New Horizons Computer Systems
- Albuquerque — Page One Newsstand
- Albuquerque — Programs Unlimited
- Albuquerque — Zia Computers
- Las Cruces — Zia Computers
- Las Cruces — Las Cruces Computers

**NEW YORK**
- Amityville — Ross Enterprises
- East Rochester — Foxx Computer Services
- Great Neck — P & T Computers
- Hicksville — Innerlogic Computer Center
- Lake Grove — Programs Unlimited
- Plattsburg — Adirondack Computer Supply
- Rochester — Computer Directions
- Rochester — Vixia Computer Center
- Wappingers Falls — Software City

**NORTH CAROLINA**
- Atlantic Beach — VBS MicroSystems
- Jacksonville — 2 M Systems
- Mt. Airy — Mountain Computers
- Raleigh — Computopia
- Raleigh — Crazy Bob's Discount Computers

**NORTH DAKOTA**
- Buxton — Systematics
- Riverside — Computer Associates

**OHIO**
- Ashland — Infopro
- Canal Fulton — Gemstone Computing
- Canton — Micro's Etc.
- Cincinnati — Cincinnati Computer Center
- Cincinnati — Video Barn
- Columbus — The Book Rack
- Euclid — Computer Warehouse
- Fairborn — Genesis Micro Systems
- Fairfield — Microwave Magic
- Kettering — Electronic Connexion
- Lombard — Complete Computing
- Mafield Heights — Programs Unlimited
- Marion — Marion Computer Center
- North Olmstead — Softwaire Centre International
- Ravenna — Alpha Computer Services, Inc.
- St. Clairsville — D & K Computer Sales & Service
- Toledo — Computer Gallery

**OKLAHOMA**
- Oklahoma City — Merit Computers
- Tulsa — Data Station — Tulsa

**OREGON**
- Beaverton — Creative Computers & Electronics
- Eugene — University of Oregon Bookstore
- Hillsboro — Quest Enterprises, Inc.
- Milwaukie — Nomies Video Tec
- Salem — Super Soft

**PENNSYLVANIA**
- Butler — Software Corner
- Downington — Downington Computer Center
- Exton — Software City
- Philadelphia — Some Hole In The Wall

**RHODE ISLAND**
- Cumberland — Mike's CP & Software

**SOUTH CAROLINA**
- Charleston Heights — Software Haus
- Travelers Rest — Concurrent Technologies Corp.

**TENNESSEE**
- Knoxville — First Byte Computer Co.

**TEXAS**
- Austin — Electronics Insight
- Houston — Computer King
- San Antonio — Wagner Bros. Computer Store

**UTAH**
- Provo — Lloyd's Business Machines

**VERMONT**
- Rutland — Computer Marketing

**VIRGINIA**
- Alexandria — Alonso Book & Periodicals
- Falls Church — Soft Cost
- Gloucester Point — Gloucester Electronics
- Norfolk — Micro-Enhancements
- Springfield — Fairfax Computer Corp.
- Virginia Beach — Beach Business Machines

**WASHINGTON**
- Bellevue — Commercial Computer Systems
- Centralia — Centralia Computer Center
- College Place — Discount Computer Supply
- Federal Way — Compu-Place
- Marysville — Crown Microproducts
- Marysville — Computer World
- Olympia — Computer Centers
- Renton — Puget Sound Electronics

| | |
|---|---|
| Seattle | Bits & Chips |
| | R&K Enterprises |
| Spokane | Quality Software |
| **WEST VIRGINIA** | |
| Charleston | Salmon's Diversified Office Services |
| Dunbar | Wilsumco |
| Morgantown | Computer Closet |
| **WISCONSIN** | |
| Janesville | Emerald Computers |
| Stevens Point | Mom's Computers |
| **AUSTRALIA:** | |
| **SYDNEY** | |
| Kingsford | Paris Radio Electronics |
| **VICTORIA** | |
| Melbourne | Computer Biz |
| **CANADA:** | |
| **ALBERTA** | |
| Calgary | Tesserac Computer Systems |
| | The Computer Shop |
| Edmonton | CompuSoft |
| | Computique |
| | Compu-ware |
| | Disk and Chip |
| | Kelly Software Distributors |
| Grand Prairie | LEW Computers |
| **BRITISH COLUMBIA** | |
| Abbottsford | Clearbrook Computer Centre |
| Richmond | Computer King |
| Penticton | Corporate Computers |
| Port Coguitlam | Compulit |
| Prince George | Computer King |
| Surrey | Computer King |
| | The Computer Dept. of London Drugs |
| Victoria | Odyssey Computer Services |
| **MANITOBA** | |
| Winnipeg | Micro-Mentor, Inc. |
| **NOVA SCOTIA** | |
| Halifax | Atlantic News |
| **ONTARIO** | |
| Burlington | Burlington Business Systems |
| Guelph | Neutron Computers, Inc. |
| Kitchener | Home Computer Centre |
| London | Town & Country Business Machines |
| Ottawa | E.T. Wilson Office Equipment LTD. |
| Peterborough | Computer Support |
| Scarborough | Scitron |
| Toronto | Computer Junction |
| | CP&A |
| Waterloo | Computer Junction |
| Windsor | Facts & Figures Computer Service |
| **QUEBEC** | |
| Montreal | Lemond PC |
| **SASKATCHEWAN** | |
| Saskatoon | Memory Lane Computers |
| **ENGLAND** | |
| East Sussex | Molimerx LTD |
| **PANAMA** | |
| Colon | Peikard, Inc. |
| **NEW ZEALAND** | |
| Palmerston North | Viscount Electronics LTD |
| **VENEZUELA** | |
| Maracaibo | Byte Computer Center |

# ADVERTISERS INDEX

We encourage you to patronize our advertisers — all of whom support the Sanyo line of personal computers. We will appreciate your mentioning **SOFT SECTOR** when you contact these firms.




Call:
Jack Garland
Garland Associates, Inc.
10 Industrial Park Road
Hingham, MA 02043
(617) 749-5852

Call:
Kate Tucci
Advertising Representative
The Falsoft Building
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

Call:
Cindy Shackleford
Director, West Coast Office
12110 Meridian South—Suite 8
P.O. Box 73-578
Puyallup, WA 98373-0578
(206) 848-7766


# Introducing the next great American success story.

The Japanese seem to have a knack for taking an existing product and improving it. Making it faster. Or more powerful. Or building in more features for the money.

Now, Sanyo has done the same thing with computers. A new line of personal computers with maximum PC compatibility.

The MBC-775 has a built-in 9″ color monitor and gives you glorious 16-color graphics capabilities.

And you'll be hard pressed to find a monochrome portable that's lighter, less bulky, less expensive with as many built-in capabilities as the MBC-675.

While Sanyo's MBC-885 family of desktop computers are among the most versatile, powerful, expandable PC-compatibles available today.

All of them give you standard features that are usually options. Like an RS-232C serial port on the MBC-675. Switchable clock rates for high speed processing on the MBC-775 and MBC-885. While every model comes with dual 360K floppy disk drives, 256K of RAM (expandable to 640K), parallel printer interface, and the capability to drive both color and monochrome monitors.

As for options, there are Sanyo's hard disk drives, expansion boards, printers and monitors. All built by Sanyo to take the hard knocks of today's business world.

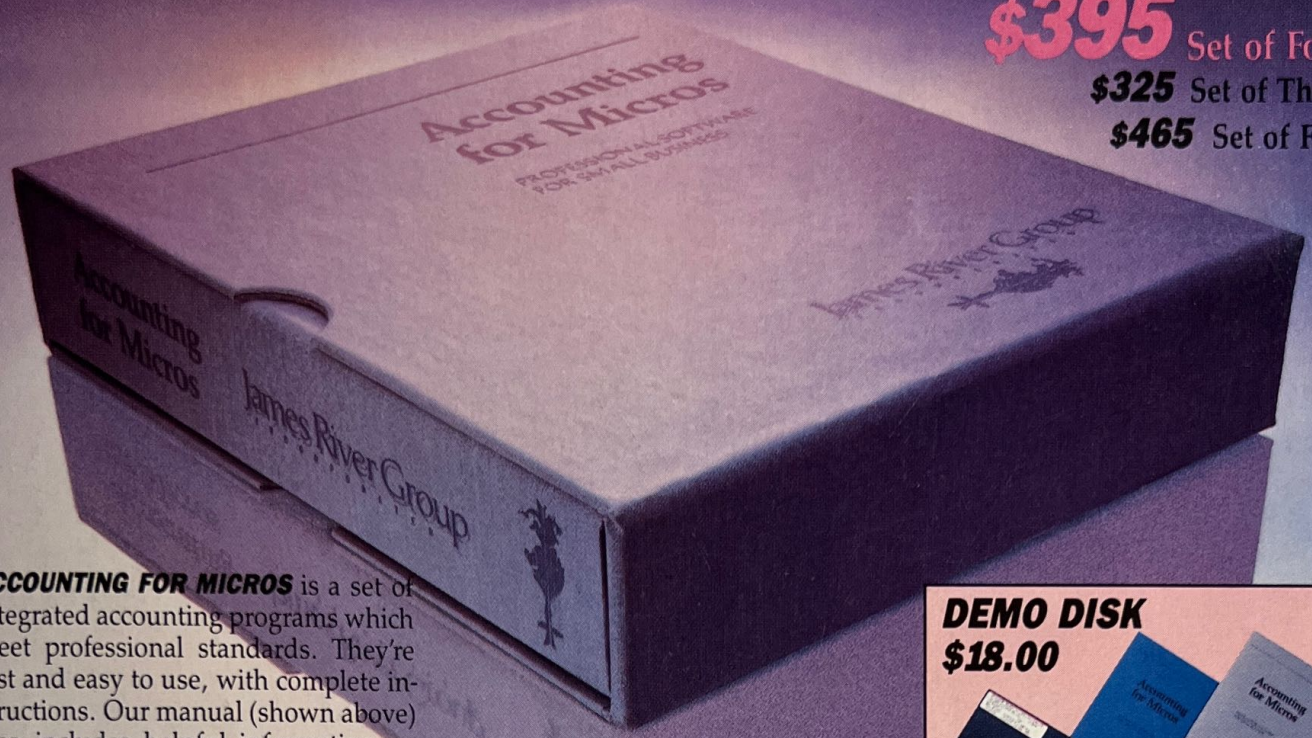
So call 1-800-524-0047 to find out where you can interface with the next great American success story: the new Sanyo line of PC-compatible personal computers.

SANYO
Sanyo Business Systems Corporation

Sanyo Business Systems Corporation, 51 Joseph Street, Moonachie, NJ 07074 (201) 440-9300

# ACCOUNTING FOR MICROS

**$395** Set of Four
**$325** Set of Three
**$465** Set of Five




***ACCOUNTING FOR MICROS*** is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

***GENERAL LEDGER ..... $125***
Allows up to 1,000 accounts & 1,000 transactions/month. Retains mo/end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

***ACCOUNTS RECEIVABLE ..... $125***
Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

***INVENTORY ..... $125***
Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

***ACCOUNTS PAYABLE ..... $125***
Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

***PAYROLL ..... $125***
Will handle up to 100 employees with eight deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately. Prints checks and W2's.

***SET OF FIVE ..... $465***
***SET OF FOUR ..... $395***
***SET OF THREE ..... $325***

***RUN ON MOST CPM AND MSDOS***

| | | |
|---|---|---|
| *Apple CPM* | *IBM PC,XT,PC jr,AT* | *Sanyo (all)* |
| *Columbia* | *Kaypro (all)* | *Tandy (all)* |
| *Compaq* | *Morrow (all)* | *TeleVideo* |
| *Corona* | *Osborne (all)* | *Zenith 100 & 150* |
| *Eagle (all)* | *Panasonic* | *8" CPM* |
| *Epson QX-10* | *Radio Shack CPM* | *Other compatibles* |

***DEMO DISK***
***$18.00***


*Try all 5 programs above (GL, AR, AP, IN, PR). Order our DEMO DISK for $18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.*

***TMAN ..... $125***
The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.
Try ***TMAN DEMO ..... $16***

***HOW TO ORDER:*** Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add $2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax). We ship most orders the same day.
or ***ORDER BY PHONE: 612-339-2521***

James River Group
INCORPORATED

(612)339-2521

125 North First Street
Minneapolis, MN 55401